DOCUMENT RESUME

ED 415 675 FL 024 905

AUTHOR	Landsbergen, Jan, Ed.; Odijk, Jan, Ed.; van Deemter, Kees, Ed.; van Zanten, Gert Veldhuijzen, Ed.
TITLE	Computational Linguistics in the Netherlands 1996. Papers from the CLIN Meeting (7th, Eindhoven, Netherlands, November 15, 1996).
INSTITUTION	Eindhoven Univ. of Tech. (Netherlands). IPO--Center for Research on User-System Interaction.
ISBN	ISBN-90-386-1051-3
PUB DATE	1997-00-00
NOTE	227p.
PUB TYPE	Books (010) -- Collected Works - General (020)
EDRS PRICE	MF01/PC10 Plus Postage.
DESCRIPTORS	*Computational Linguistics; *Computer Software; Foreign Countries; *Language Patterns; Language Research; Linguistic Theory; Polish; *Programming; Research Methodology; *Speech Synthesizers; Uncommonly Taught Languages
IDENTIFIERS	Netherlands

ABSTRACT

Papers from the meeting on computational linguistics include: "Conversational Games, Belief Revision and Bayesian Networks" (Stephen G. Pulman); "Valence Alternation without Lexical Rules" (Gosse Bouma); "Filtering Left Dislocation Chains in Parsing Categorical Grammar" (Crit Cremers, Maarten Hijzelendoorn); "Speech Output Generation in GoalGetter" (Esther Klabbers); "Possessive Affixes and Complement Composition" (Dimitra Kolliakou); "Presuppositions as Anaphors Revisited" (Emiel Krahmer, Kees van Deemter); "Modeling Coordination by Means of Operations on Strings and on Derivation Trees" (Carlos Martin-Vide, Georghe Paun); "Improving the Precision of a Text Retrieval System with Compound Analysis" (Renee Pohlmann, Wessel Kraaij); "Unbounded Negative Concord in Polish: A Lexicalist HPSG Approach" (Adam Przepiorkowski, Anna Kupsc); "Information Update in Dutch Information Dialogues" (Mieke Rats); "ANNO: A Multi-Functional Flemish Text Corpus" (Ineke Schuurman); "GoalGetter: Predicting Contrastive Accent in Data-to-Speech Generation" (Mariet Theune); "On the Notion 'Minor Category'" (Frank van Eynde); and "Resolving PP Attachment Ambiguities with Memory-Based Learning" (Jakub Zavrel, Walter Daelemans, Jorn Veenstra). (MSE)

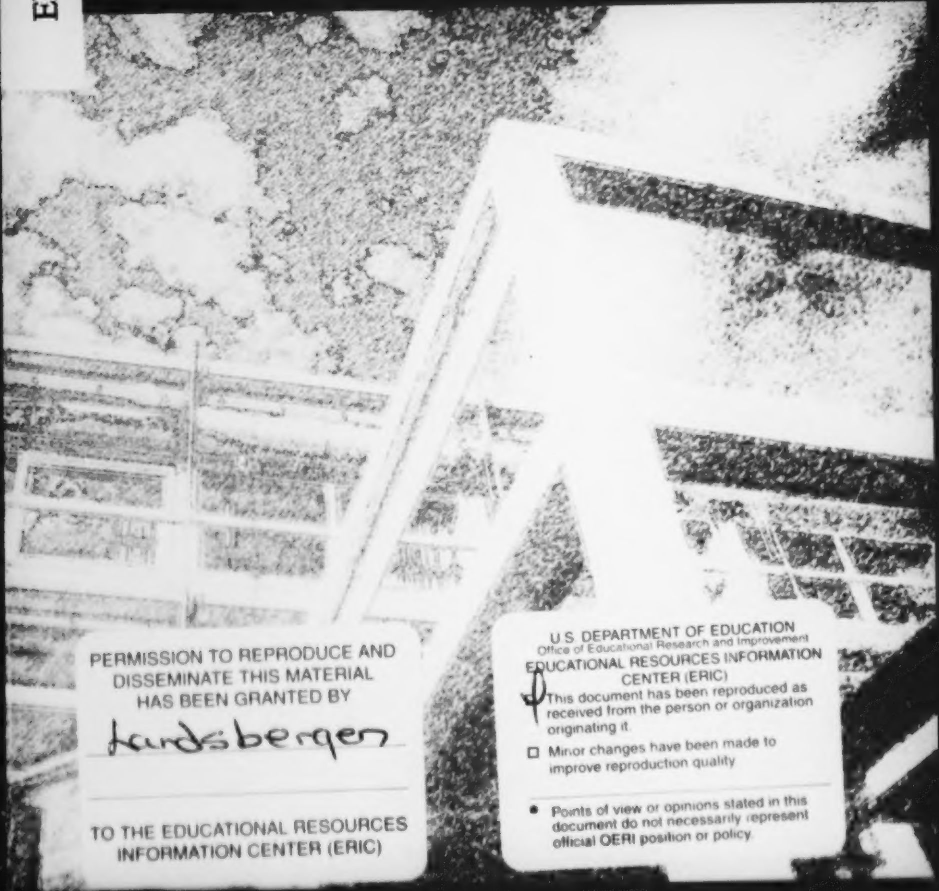
****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 415 675

IPO Center for Research
on User-System Interaction

Computational Linguistics in the Netherlands 1996

Papers from the Seventh CLIN Meeting



PERMISSION TO REPRODUCE AND
DISSEMINATE THIS MATERIAL
HAS BEEN GRANTED BY

Landsbergen

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

● Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Jan Landsbergen, Jan Odijk, Kees van Deemter
and Gert Veldhuijzen van Zanten (eds.)

tu

2 BEST COPY AVAILABLE

COMPUTATIONAL LINGUISTICS IN THE NETHERLANDS 1996

Papers from the Seventh CLIN Meeting

Jan Landsbergen,
Jan Odijk,
Kees van Deemter and
Gert Veldhuijzen van Zanten (eds.)

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Computational

Computational Linguistics in the Netherlands:
Papers from the Seventh CLIN Meeting /
Jan Landsbergen, Jan Odijk, Kees van Deemter and
Gert Veldhuijzen van Zanten (eds.). - Eindhoven :
IPO, Center for Research on User-System Interaction. 1997
ISBN 90-386-1051-3
Subject heading: Computational Linguistics

Preface

This book contains a selection of papers presented at the seventh CLIN (Computational Linguistics in the Netherlands) meeting. The meeting was held on November 15, 1996, at IPO, on the premises of the Eindhoven University of Technology.

The aim of the annual CLIN meetings is to provide an opportunity for computational linguists to report on their work. The CLIN meeting also functions as an informal meeting place, primarily for Dutch and Belgian computational linguists, but with an increasing international participation. We were especially happy that Stephen Pulman (SRI International and University of Cambridge) was willing to act as our keynote speaker.

About 70 participants attended the meeting, the program listed 24 presentations, of which 16 were submitted for inclusion in the proceedings. After the reviewing procedure 13 papers remained, which you will find in this book, preceded by Pulman's invited paper. The contents of the CLIN proceedings will also be made available electronically, via World Wide Web, as a subpage of CLIN's Home Page:

`http://odur.let.rug.nl/~vannoord/clin/clin.html`

We would like to thank all those who contributed to CLIN VII: the speakers, the participants, the external reviewers (Gosse Bouma, Walter Daelemans, Frank van Eynde, Theo Janssen, Anton Nijholt, Remko Scha and Gertjan van Noord) and the people of IPO's service department. We also thank IPO and the NWO Priority Programme on Language and Speech Technology for sponsoring the meeting.

CLIN VII was held in a period that IPO went through a drastic transformation process, which unfortunately resulted in the dissolution of the Language Group, to which the organizers belonged. We thank our CLIN colleagues for their moral support.

The eighth CLIN meeting will be held in Nijmegen on December 12, 1997. We are looking forward to seeing you there.

Eindhoven, October 1997
Jan Landsbergen, Jan Odijk, Kees van Deemter and Gert Veldhuijzen van Zanten.

Author Index

Gosse Bouma 25
Crit Cremers.................... 41
Walter Daelemans..............207
Maarten Hijzelendoorn......... 41
Esther Klabbers................ 57
Dimitra Kolliakou.............. 69
Wessel Kraaij..................115
Emiel Krahmer................ 85
Anna Kupść....................129
Carlos Martín-Vide............101
Gheorghe Păun.................101
Renée Pohlmann...............115
Adam Przepiórkowski..........129
Stephen G. Pulman............ 1
Mieke Rats.....................145
Ineke Schuurman...............161
Mariët Theune.................177
Kees van Deemter.............. 85
Frank van Eynde...............191
Jorn Veenstra..................207
Jakub Zavrel....................207

6

Table of Contents

Preface iii

Author Index iv

Invited paper:

Stephen G. Pulman
Conversational Games, Belief Revision and Bayesian Networks 1

Gosse Bouma
Valence Alternation without Lexical Rules 25

Crit Cremers and Maarten Hijzelendoorn
Filtering Left Dislocation Chains in Parsing Categorial Grammer 41

Esther Klabbers
Speech Output Generation in GoalGetter 57

Dimitra Kolliakou
Possessive Affixes and Complement Composition 69

Emiel Krahmer and Kees van Deemter
Presuppositions as Anaphors Revisited 85

Carlos Martín-Vide and Georghe Păun
Modeling Coordination by Means of Operations on Strings and on Derivation Trees 101

Renée Pohlmann and Wessel Kraaij
Improving the Precision of a Text Retrieval System with Compound Analysis 115

Adam Przepiórkowski and Anna Kupść
Unbounded Negative Concord in Polish: A Lexicalist HPSG Approach 129

Mieke Rats
Information Update in Dutch Information Dialogues 145

Ineke Schuurman
ANNO: a Multi-functional Flemish Text Corpus 161

Mariët Theune
GoalGetter: Predicting Contrastive Accent in Data-to-Speech Generation 177

Frank van Eynde
On the Notion 'Minor Category' 191

Jakub Zavrel, Walter Daelemans and Jorn Veenstra
Resolving PP attachment Ambiguities with Memory-Based Learning 207

Addresses of Authors 223

Invited paper:

Conversational Games, Belief Revision and Bayesian Networks

Stephen G. Pulman*

Abstract

The paper uses a simple and abstract characterization of dialogue in terms of mental state changes of dialogue participants to raise three fundamental questions for any theory of dialogue. It goes on to discuss currently popular accounts of dialogue with respect to these three questions. Next, the notion of 'conversational game' is revisited within a probabilistic and decision theoretic framework, and it is argued that such an interpretation is plausible both intuitively and as the basis for computational implementation. An illustrated sketch of a proposed implementation using Bayesian networks is described.

Three Questions for Dialogue

A simple, rather abstract description of a canonical dialogue is that it consists of a sequence of utterances with a corresponding sequence of mental states of the participants in the dialogue. Person A has a sequence of mental states $S_{A1} \ldots S_{An+1}$ and person B also has a sequence $S_{B1} \ldots S_{Bn+1}$. Connecting these two sequences is a third sequence, the sequence of utterances. U_{A1} is produced by A in state A1, U_{B2} is produced by B in B2 and so on. Furthermore, A's state S_{A2} and B's state S_{B2} are, at least partially, determined by the utterance U_{A1} which precedes them. The utterances change the mental states of the participants to the point where no further communication is regarded by them as necessary: the goals of the conversation, whatever they were, have been achieved as far as is possible. This is represented by the diagram in figure 1.

Even this simple picture reveals that there are several large questions to be answered in order to be in a position to build a machine capable of playing the part of A or B:

(i) what are mental states?

(ii) how do they change?

(iii) how do utterances connect with them and change them?

---

*SRI International Cambridge Computer Science Research Centre and University of Cambridge Computer Laboratory.

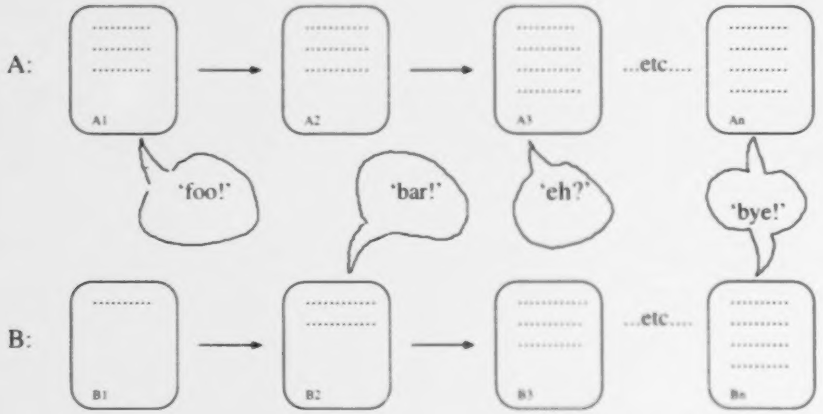


Figure 1: Two-person Dialogue

1 The BDI tradition

Insofar as the current literature on computational models of dialogue has a received wisdom on the answers to these questions, it is probably that given by the 'BDI' model of rational agency, as described for example in Cohen, Morgan, and Pollack (1990). The answer to the first question is that mental states are, or can be modelled as, sets of sentences in some logic, expressing the Beliefs, Desires, and Intentions of an agent (see Cohen and Levesque (1990)). Various axioms connect the having of desires and intentions with the performance of actions, some of which are linguistic actions. A rational agent, given an initial mental state, will reason as to the best course of action so as to fulfil the highest priority desires. Conversation proceeds via the performance of these linguistic actions. Part of the reasoning involves a model of the mental state of the other participants, and inferences about what their goals and intentions might be, based on the observed linguistic acts they carry out.

For a partial answer to the second question, how do mental states change, if mental states are modelled as sets of sentences in some logic, then it is appropriate to turn to the belief revision literature: e.g. Gärdenfors (1988), Galliers (1990). Belief revision is modelled via the addition or subtraction of propositions (if expressed on closures of belief bases, i.e. the deductive closure of some set of axioms) or of sentences (if expressed on belief bases themselves), operations which are required to preserve consistency. It is in the latter sentential form in which belief revision has to be implemented for the purposes of computational dialogue modelling, of course. A simple approach to belief revision within this framework would posit two basic operations, given a set of sentences Δ representing the existing mental state, and a sentence α, which is some component to be added or removed as the result of processing an utterance.

Subtraction:

If Δ does not entail α then $\Delta'=\Delta$;

Else, find some β in Δ such that $\Delta - \beta$ does not entail α,
and $\Delta' = \Delta - \beta$

In many cases β and α will be the same, or α will follow directly from β perhaps in conjunction with some other sentences which taken alone do not entail α. Of course β may be a conjunction of several different sentences.

Addition:

If Δ does not entail $\neg\alpha$, then $\Delta' = \Delta + \alpha$;

Else, find some β such that $\Delta - \beta$ does not entail $\neg\alpha$,
and $\Delta' = (\Delta - \beta)+\alpha$

It is worth noting that we need not just belief revision, but also revision of desires, and intentions. Conflicting goals and incompatible intentions are drivers of conversational processes just as much as detection of mismatches in beliefs. It is also worth pointing out that the mechanisms presupposed in belief revision, like detection of inconsistency or conflict, are required in some form for approaches that do not necessarily describe themselves as doing belief revision. Any approach to dialogue needs to be able to tell when an answer to a question is a plausible and appropriate one; when two goals cannot both be simultaneously achieved; or when some piece of information is implied by what is mutually known and therefore need not be explicitly repeated. Any formal mechanism that achieves this is addressing the problem of belief revision.

Let us turn now to the answer given to question (iii), how do utterances relate to, and change, mental states? The BDI answer to this question is essentially that derived from the speech act literature, as presented by Cohen and Perrault (1979) and Perrault and Allen (1980). Characterising an utterance as a particular type of speech act enables it to be related to properties of the speaker's mental state, by linguistic and other conventions governing that type of act. These conventions ('felicity conditions' in the original formulation) represent necessary and sufficient conditions for the performance of a genuine instance of a particular kind of speech act as in Searle (1969), and those conditions are at least in part conditions on the speaker's mental state, requiring the speaker to have the right kind of beliefs, desires, and intentions. Thus a hearer can make inferences about the speaker's mental state once an utterance has been recognised as instantiating a particular kind of speech act.

Given background axioms of 'rational agency' characterising the behaviour of an ideally cooperative rational hearer, the BDI approach also has an account of how an utterance can change the mental states of the participants in a dialogue. As an illustration of the general approach, a typical story about how a request can lead to a change of mental state and a consequent action on the part of a hearer will go something like this. We assume that the speech act conditions, and the rational agency axioms are characterised along roughly these lines:

```
Request Precondition:
IF Speaker wants A
   AND Speaker believes Hearer can do A
   AND ... etc.
THEN Speaker requests Hearer to do A

Request Postcondition:
IF Speaker requests Hearer to do A
   AND ... etc.
THEN Hearer believes Speaker wants A
```

Axioms of 'rational behaviour':

```
Cooperativity:
  IF Hearer believes Speaker wants A
     AND ... etc.
  THEN Hearer wants A

Desire leads to Action:
  IF X wants A
     AND X can do A
  THEN X does A
```

Now a typical piece of reasoning that could lead a Speaker to make a request in order to achieve some desire might proceed as follows:

Speaker requests Hearer to do A	
∴ Hearer believes Speaker wants A	(Request Postcondition)
∴ Hearer wants to do A	(Cooperativity)
∴ Hearer does A	(Desire leads to Action)

That is, the Speaker desires that A be done and he reasons that by issuing a request he will start the above chain of events that results in A being done. This reasoning is typically done by backward chaining from the goal state, but that is really an implementational issue that does not affect the logic.

1.1 Some problems for the BDI tradition

The BDI tradition has led to many theoretical insights into the nature and functioning of dialogue, and there are several very impressive implemented systems based on versions of the approach: for example, those described by Allen, Miller, Ringger, and Sikorski (1996) or Sadek, Ferrieux, Cozannet, Bretier, Panaget, and Simonin (1996). Nevertheless there are several areas where the theoretical content is unclear or questionable, and there are many aspects of the theory which do not seem likely to yield satisfactory large scale computational implementations. We turn now to discussion of some of these problems.

The basic propositional attitudes countenanced by the BDI tradition are those from which it derives its acronym: belief, desire, and intention. However, since the earliest formal work on dialogue it has been recognised that many of the propositions that correspond to utterances in a dialogue do not fall easily into these three categories. Hamblin (1971) pointed out (p 36ff) that many sentences correspond to propositions that are not (yet, anyway) believed by the participants. He introduces the notion of a commitment, which is not necessarily a belief (though it may become one) but a function purely of what has been said. Speakers are generally committed to a statement if they make it, or agree to one made by someone else, or if it clearly follows from something else to which they are committed. In particular commitments may be later *retracted* but not *denied*.

In the recent literature other closely related terms have been used. Traum uses the term 'proposal' in Traum and Hinkelman (1992) and the idea of propositions that are being 'grounded' but not yet agreed appears in Clark and Schaefer (1989). For example, in the following dialogue (from the 'Autoroute' corpus described by Moore and Browning (1992)), between a 'wizard' pretending to be a route-planning system, and a caller, the proposition 'caller wants to go to Edwinstowe' cannot be said to be a belief of the wizard until at least step 4, rather than step 2, where the proposition is 'in the air'. (We assume throughout that what is happening is that at step 3 the wizard is not sure she has heard correctly. At step 6 the system she is operating has reported that there is more than one Edwinstowe).

```
1. w: Where would you like to go?
2. c: Edwinstowe
3. w: Edwinstowe?
4. c: Yes
5. w: Please wait
6. w: Is that Edwinstowe in Nottingham?
7. c: Yes
```

More recently, several authors, like Traum and Allen (1994) and Bunt (1997), have pointed to the need to also recognise a category of 'obligations' or 'social commitments' which arise from linguistic and social conventions. If someone asks you a question, you are, as a reasonable member of the same language community, thereby placed under some kind of obligation to respond.

Many other types of phenomena that are encountered in real dialogues seem to resist an easy classification into one of the three propositional attitudes countenanced by the approach. These include what Bunt calls 'dialogue control' phenomena: utterances (feedback, acknowledgements, pause-fillers, etc.) whose function is to maintain the dialogue and coordinate the participants, rather than to directly express beliefs, desires, or intentions.

These observations do not threaten the central role of beliefs, desires and intentions, of course, but they do indicate that as an empirically adequate account of what actually goes on in dialogues the BDI approach needs considerable supplementation and extension. The notion of 'mental state' provided by the theory is too simple to explain everything that happens in a natural dialogue.

Let us turn now to the question of change of mental state, and the belief revision framework assumed implicitly or explicitly by BDI approaches.

The classical belief revision framework (and associated approaches such as dynamic logic: Groenendijk and Stokhof (1991), Jaspars (1996)), while giving a clear logical theory of change of information state, present many problems when large scale practical implementations are contemplated. As is well known, a simple method of belief revision like that sketched above is very highly non-deterministic. Even given such simple choices for the existing set of beliefs Δ and a candidate for addition or subtraction α as:

$$\Delta = \{a, a \rightarrow b\}, \ \alpha = b \text{ (Subtraction) or } \neg b \text{ (Addition)}$$

there will be a choice about which β to remove. Practical belief revision requires us to assume some priority ordering on sentences in a belief base, such that given several candidates for elimination, the one which is 'cheapest' in terms of some overall score will be given up. This priority ordering usually corresponds to an intuitive notion like 'strength of belief' or 'degree of commitment'. Deciding on the adjustment that makes the least overall change required to preserve consistency can be a computationally intensive operation. Note that any such system of weighting is not part of the logic itself and so some separate mechanism is required to make sure that the weighting scheme itself observes reasonable properties.

Implementing classical belief revision of course requires us to be able to detect inconsistency, and thus some kind of classical negation is necessary in our logics. It would be impossible to do belief revision on sets of pure Horn clauses, for example. But this means that we have problems, not just with efficiency, but also with 'logical omniscience'. If the logic is strong enough to detect inconsistencies between complex beliefs, it is likely also to make the contents of a belief state imply logical consequences of basic beliefs that are actually beyond human ability to compute.

For both of these reasons it is desirable for an implementation also to model something like 'focus of attention' or 'salience' of sentences in the mental state, so that reasoning can be restricted to relevant subsets of sentences, and conclusions can be limited to those that are humanly processable. However, all the obvious ways of achieving this notion (e.g. limiting chains of inference to a certain depth) compromise completeness and (global) consistency, as discussed in Konolige (1986). Since these are not properties that characterise human reasoning, especially in dialogue, this may actually turn out to be an advantage to us, but nevertheless it is not easy to see how to achieve exactly the right kind of restrictions without unwanted negative effects.

Lastly, but by no means least, there is the fact that the classical approach to belief revision requires us to axiomatise the relevant properties of the domain in order to be able to track what follows from what. As anyone who has ever tried to carry out such an exercise in knowledge representation will confirm, this is an exceedingly difficult undertaking, especially when classical first order logic is the representation language. It soon becomes obvious why all the textbook examples are simple blocks worlds, or equally well structured and clean domains. Anything else is generally just too messy and hard, and the resulting axiom set is always very fragile and incomplete in its coverage.

Turning now to the third of our questions, how to connect utterances with mental state, we also find problems with the logical reconstruction of speech act theory that is needed within the BDI framework. For example, many people, not least the original proponents of the theory, have commented on the implausibility of the 'Cooperativity' axiom (and its analogues for other speech acts). There are actually two problems: firstly, it does not allow for the case where the hearer might not want to cooperate, or where external circumstances may bring about conflicting goals if he does: see Galliers (1990). It can also be the case that a hearer might be cooperative in some respects but not others. To some extent this can be alleviated by introducing some notion of defaults (although how to square this with the requirements of classical belief revision is not obvious).

Secondly, and more seriously for the interpretation of the BDI account as a contribution to a theory of *dialogue*, is the fact that these axioms are, in the theory, the only way of achieving 'uptake' of a speech act; that is, of creating a link between an utterance by a speaker, and subsequent modification of the hearer's beliefs or intentions concerning anything other than the speaker's mental states. In many respects, the original speech act theory is rather solipsistic or one-sided: it deals with the conditions for the successful performance of some act by a speaker, but has virtually nothing to say about what happens next, or in fact about anything outside the speaker's head. For example, as far as speech act theory proper is concerned, it is largely unexplained why a request is typically met either with an acceptance or a refusal, or why a question is typically met with an answer rather than (say) a request or another question. In the speech act literature, and in the BDI tradition derived from it, there are no dialogue units larger than a single utterance: a response to a request, or an answer to a question, cannot within the theory be distinguished from a conversation-initiating declarative.

Also completely unexplained, even with the appropriate axioms in place, is why there is a pressure on a hearer to respond somehow to an utterance even if he is not in a position to respond appropriately to it. Requests which are not going to be complied with are still acknowledged; questions that cannot or will not be answered still evoke some kind of explanation or diversion. Complete silence is not an option, although it is not easy to see how that option would conflict with anything in speech act theory.

2 Responses

There have been broadly two types of response to this problem. (Actually, only one is a direct response; the other is more of a parallel development that can also be seen as offering a solution). Traum and Allen (1994) propose the addition of a new mechanism to a speech act-based approach, namely 'discourse obligations'. A discourse obligation is a linguistically based social convention having the effect that when a particular speech act is recognised by a hearer, the hearer incurs an obligation to respond in an appropriate way:

Speech Event	Discourse Obligation
S request A	H accept or reject A
S ask whether P	H say whether or not P
Utterance failure	H repair utterance
etc.	

Thus we now have what might be called a BDIO model: a new propositional attitude is added. However, the notion of a 'speech event' is now much wider than that of a speech act: although the latter, in their original formulation at least, e.g. in Searle (1969), included some acts that one might think of as dialogue control acts rather than as BDI related. Nevertheless, even in the most ambitious formulations, there was no speech act of utterance failure.

However, while this formulation begins to describe the conventional association between questions and replies, requests and acknowledgements, and so on, it does not fully capture the nature of the more general social pressure to respond that is characteristic of normal dialogues. For example, in cases where a politician is asked an awkward question in an interview, he will usually fail to obey the specific question-related discourse obligation described above, but he cannot just remain silent. What he will typically do is talk about something else that he hopes will be taken as a relevant response, but which does not actually constitute an answer. It seems plausible that there are at least two types of obligation involved in discourse: those which are associated with particular speech acts or utterance types (e.g. that a yes/no question demands the answer yes or no), as described by Traum and Allen, and those which are more general social and communicative obligations, not specific to particular constructs, and concerned with the maintenance of communication norms.

The second line of work which can be seen as addressing this particular defect of speech act theory is the 'Conversational Games' tradition: Power (1979), Houghton (1986), Kowtko, Isard, and Doherty (1992), Reithinger and Maier (1996). More of a descriptive framework than a theory, this tradition posits a set of 'conversational games' or 'dialogue games' each consisting of a set of moves, where an utterance may realise one or more moves. The important thing is that the games encompass both partners in dialogue: for example, a yes/no game consists of a yes/no question along with its yes/no reply. Thus the conventional link between utterance type and response type is achieved by making the unit of discourse something that by definition is not restricted to a single utterance. This may not be a very sophisticated theoretical innovation, but it at least describes the facts correctly.

Some conversational games postulated by Kowtko, Isard, and Doherty (1992) on the basis of study of the Edinburgh 'map task' corpus are: Instruction, Confirmation, Question-YN, Question-WH, Explanation, Alignment. The moves can be broken into two categories:

Initiating Moves:

`Instruct`	`(provides instruction)`
`Check`	`(elicits confirmation of known information)`
`Query-yn`	`(asks yes-no question for unknown information)`
`Query-wh`	`(asks wh-question for unknown information)`
`Explain`	`(Gives unelicited description)`
`Align`	`(Checks alignment of position in task)`

Response and feedback moves:

`Clarify`	`(clarifies or rephrases given information)`
`Reply-y`	`(responds affirmatively)`
`Reply-n`	`(negatively)`
`Reply-wh`	`(Respond with requested information)`

Acknowledge	**(acknowledge and request continuation)**
Ready	**(Indicates intention to begin a new game)**

In principle the framework of conversational games can easily cover those utterance types that do not fit happily into a pure speech act framework, recognising that the function of some of these is to provide information about the state of the dialogue (e.g. alignment - making sure both partners know where they are in the dialogue) and to increase the degree of confirmation about some piece of information.

A more refined characterisation of these 'dialogue control' acts is given by Bunt (1997). He distinguishes different aspects of context: semantic, cognitive, physical, social, and linguistic, with different types of dialogue act for each. Dialogue acts are acts which change one or more aspects of the context.

Conversational Games are a useful descriptive framework. But as a theoretical contribution to the understanding of dialogue, they have remained somewhat weak. Firstly, it is not clear how they differ from the BDI framework in the way they try to establish a link between utterances and mental states. From the perspective of speech act theorists, conversational games look like a hard-wiring of some of the patterns of inference that they derive from first principles.

Secondly, the theory seems very unconstrained. For example, is there a satisfactory answer to the questions of how many games there are, how they vary according to the type of dialogue, and what constraints there are upon possible games? These are the kinds of question routinely asked of every other level of linguistic formalism. For example, in the Verbmobil system as described in Reithinger and Maier (1996), games of much finer level of detail than in the Map Task are envisaged: e.g. 'arranging a time', or 'confirming a date'. These are justified along exactly the same lines as those developed for the Map task, namely, intuitive agreement that a certain level of commonality exists between different utterance/context pairs. But it is clearly not very much further down this route before there is a distinct game for practically every utterance. We would therefore like some theoretical grounding to establish what granularity is characteristic of useful games.

To illustrate these issues, consider the question: What distinguishes a move from a game? One cannot simply identify games with standardised sequences of moves, although this is at first sight a tempting idea (and explicitly proposed in Houghton (1986)). For example, one might think that a WH query game should consist of a WH-query move followed by a WH-reply move. But if this were the case then we would have to say that the WH query game in the dialogue fragment we saw earlier would be over after turn 2, whereas intuitively one would want to say that it was only completed after the two checking games.

		Move	**Game**
1. `w: Where would you like to go?`		query whq	WH
2. `c: Edwinstowe`		reply whq	
3. `w: Edwinstowe?`		check	CHK
4. `c: Yes`		clarify	
5. `w: Please wait`	(time management)	align/acknowledge	
6. `w: Is that Edwinstowe in Nottingham?`		query ynq/check?	CHK
7. `c: Yes`		reply yes/clarify	

Ian Lewin has suggested (p.c.) that in general we should single out those (sequences of) dialogue acts that serve to change the status of propositions currently under discussion from 'proposals' to 'agreed commitments'. The boundaries marked by these transitions do seem, at least in these types of dialogue, to correspond to natural divisions in a dialogue. Thus although there is a context change between each of the utterances above, and there are three games played, there is only one significant change to the agreed commitments of the participants. Utterances 4-7 serve to check and ground the information introduced by 1 and 2, and so although they do change the linguistic and other aspects of the context, there is a good sense in which they have a different status. Lewin points out that it is plausible, for example, that to the extent that dialogues are consciously or unconsciously planned, the units of planning are those represented by the acquisition of agreed propositions rather than the units that correspond to the conversational games like 'checking' or 'acknowledgement'. It is not plausible to assume that such moves are planned: rather, they arise as an immediate response to the current state of the dialogue.

3 Conversational Games Reconstrued

Let us reconsider what a notion of conversational game might tell us about the answers to the three questions with which we began our investigation. In particular, we will explore a somewhat different, and in some ways more traditional, interpretation of the notion of a 'game'.

We consider (task-oriented) dialogues to be a kind of game whose goal is to achieve the purposes of the dialogue (e.g. booking an airline ticket, planning a car journey) usually as quickly and economically as possible. A suitable example game to explain the analogy might be a card game like bridge. The players are in the position that a certain amount of information about the hand that the other player has is overtly available via the content of utterances, but the rest has to be inferred on the basis of bid behaviour and knowledge about cards. Some good reasoning or lucky guesses may lead to a speedy conclusion of the game. But a bad guess might put one at a disadvantage. So each move has to be made with an eye to its possible positive or negative effects. In formal decision theory, the effects are of course called 'utilities', and each move is calculated (if the player is rational) to maximise utilities. Moves are seldom made simply in response to the previous move by the opponent (although sometimes this is necessary, as when to move a king out of check) but are more often part of a longer range strategy.

Pursuing the analogy at a more detailed level, then, our conversational game framework requires at least the following components:

(i) move interpretation: when a player puts down some cards, we use that information to work out what other cards the player may have, or may want. The conversational game analogue of this to the classification of an utterance as a realisation of one or more conversational moves. Classifying an utterance as a move is making one hypothesis about the speaker's mental state. Equally, one may make further hypotheses about what it is reasonable to think led to that particular move being made.

(ii) tactics: planning the next move in response both to the immediate situation but also the longer range strategy. In some cases the immediate situation may be the most important factor, as when moving a piece to avoid capture, or requesting clarification when an utterance has not been recognised with sufficient confidence, or when it presents the belief revision component with an apparent contradiction. But if things are going according to plan, the next move is both an appropriate response to the previous one, and a step forward in the overall plan.

(iii) strategy: planning the next game or sequence of games to be played in order to win the dialogue. Strategy needs to be continually re-evaluated as new information is obtained.

(iv) a fourth but vital component is that knowledge of the domain which supports the various types of reasoning in (i-iii).

As the computational underpinning of all four components we intend to explore the use of Bayesian Networks, as developed by Pearl (1988), and described in Neapolitan (1990), a formalism which is becoming widely used in the AI community for knowledge representation, causal reasoning, belief revision, and decision theoretic reasoning. There is little doubt that, modulo some important provisos below, this formalism can provide a plausible platform for component (iv) and so in the remainder of this section we concentrate on (i)-(iii).

4 What is a Bayesian Network?

Given a probability space of events, E, a 'propositional variable' is a function from E to a finite subset of E of mutually exclusive and exhaustive events. Given a propositional variable A, let $a_1 \ldots a_n$ be the set of possible values of A. We write $P(A = a_i)$ as $P(a_i)$ and an expression like $P(A \mid B) = P(A)$ is a shorthand for the expressions $P(a_i \mid b_j) = P(a_i)$ for all i and j. Given a set of propositional variables A, B, C, ... we can define a joint probability distribution on them such that:

$$\sum_{ijk\ldots} P(a_i, b_j, c_k, \ldots) = 1$$

Given a set of such variables, $\{X_1 \ldots X_n\}$, the 'marginal probability' of any subset of them, say $X_i, \ldots, X_j$, relative to this joint probability distribution is defined as:

$$P(X_i, \ldots, X_j) = \sum_{k \neq i \ldots j} P(X_1 \ldots X_n)$$

A Bayesian or causal network is a set of propositional variables, associated with vertices in a directed acyclic graph, where there are conditional (in)dependencies between some of the variables, as reflected pictorially in the associated graph. More formally, a DAG consisting of vertices/variables V and edges E, with an associated joint distribution P, constitutes a Bayesian network under conditions below.

First, given a variable v which is a member of V, let $c(v)$ ('causes of v') be the set of v's parents, let $d(v)$ be the set of v's descendants, and let $a(v)$ be $V - (d(v) \bigcup v)$, that is, all the variables except v and v's descendants.

Let W be any subset of $a(v)$. W and v are conditionally independent given $c(v)$, where $P(c(v)) \neq 0$, under three conditions:

if $P(v \mid c(v)) = 0$
- because nothing further (in particular W) can affect v

if $P(W \mid c(v)) = 0$
- because nothing further (in particular v) can affect W

if $P(v \mid W \bigcup c(v)) = P(v \mid c(v))$
- which can be verified by calculation

If every subset of W is conditionally independent of v given $c(v)$ then the DAG is a Bayesian network.

The thing to notice about conditional independency is that although some independencies will be permanent because of the configuration of the network, and will not be affected by instantiation of variables (i.e. when it is known which value of the propositional variable = 1), some variables will become independent of each other only when some intervening variable has been instantiated.

The conditional independencies in a network can be exploited to reduce the amount of computation involved in working out joint probabilities. Take for example, a network of the form

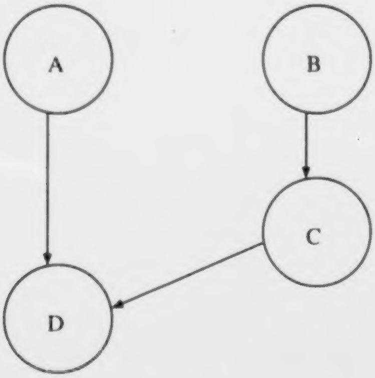


Figure 2: bayesian net

The 'chain rule' of probability theory tells us that the joint probability can be calculated from conditional probabilities thus:

$$P(A, B, C, D) = P(A \mid B, C, D) \times P(B \mid C, D) \times P(C \mid D) \times P(D)$$

In order to fully exploit this equivalence we must reorder the variables so as to reflect the structure of the DAG.

$$P(D,C,A,B) = P(D \mid A,C,B) \times P(C \mid A,B) \times P(A \mid B) \times P(B)$$

Now we can use the structure of the DAG to determine the conditional independencies: A and B have no parents so they are not dependent on any other variable: thus $P(A \mid B) = P(A)$. Variable C is dependent only on the value of B so $P(C \mid A,B) = P(C \mid B)$. Variable D is dependent only on the value of A and of C (since any effect of B has to be via C) and so $P(D \mid C,A,B) = P(D \mid C,A)$.

$$P(D,C,A,B) = P(D \mid A,C) \times P(C \mid B) \times P(A) \times P(B)$$

For larger networks, this simplification avoids many unnecessary computations.

Now the general version of the chain rule for Bayesian networks can be written:

$$\prod_i P(v_i \mid c(v_i)) where P(c(v_i)) \neq 0$$

This enables us to compute the joint distribution from the conditional probabilities. We can also compute the conditional probabilities given the joint distribution, using the chain rule in the other direction:

$$P(A \mid B,C,D) = \frac{P(A,B,C,D)}{P(B,C,D)}$$

and so on.

When a variable (usually one with no parents or no children) is instantiated, i.e. when we know which of its values is the observed one, the probabilities in the network have to be updated. This is done by propagation from the instantiated variable. The probability of each variable V can be calculated by combining the evidence for V from the nodes above it in the network, and those from below: let the evidence from the parents of V be E_c and the evidence from the daughters of V be E_d. Then:

$$P(V \mid E_c, E_d) = \frac{P(E_d \mid V) \times P(V \mid E_c)}{\alpha}$$

where α is a normalising constant.

The precise algorithm for computing these quantities and for propagating their effects throughout the relevant portions of the network is very complex, since a node may have many parents and many children and allowance has to be made for the mutual effect of new information on any of these. The algorithm assumes that networks are 'singly connected' i.e. that for any pair of nodes there is only one path that can be found between them (ignoring directions on arcs). This is not a limitation in principle because any multiply connected graph can be transformed to a singly connected one, although at some resulting computational cost.

The original version of the algorithm can be found in Pearl (1988); a very detailed tutorial description can be found in Neapolitan (1990); and a simplified version restricted to tree-shaped networks is given in Shoham (1994).

One serious restriction that Bayesian networks impose is that the variables are propositional: i.e. they have only a finite number of atomic values. This means, in effect, that quantificational reasoning, or reasoning that depends on the internal structure of propositions, is not directly possibly. However, the networks can be very large: many applications use networks of tens of thousands of nodes each with a large number of values, and so for many practical purposes this restriction does not begin to bite. Propositions of the form 'pred(A,B)' can be modelled as a node 'pred' with values in the product A × B of all relevant A and B values. Implementational devices can be used to keep this kind of thing manageable. An alternative is to generalise a potentially infinite number of propositions to a 'proposition type' which stands for all of them, if the differences between tokens are not important.

5 Bayesian networks for move recognition

When a hearer categorises an utterance as realising a conversational move there are presumably several factors taken into account in making this decision. Firstly, the linguistic form and content of the utterance is important: for example, it is very unusual for an utterance of 'no' to be interpreted as realising a 'reply-y' move, (although possible if enough intonational cues are given to signal a non-literal interpretation). Secondly, the previous few moves, or perhaps the recognition of the game currently being played has to play an important part. (In some approaches it is the only factor taken into account: see Reithinger and Maier (1996)). Thus an utterance of 'OK' might be interpreted as a 'reply-y' move if the previous move was a 'query-yn', but if the previous game has been seen to be completed it is more likely to be a 'ready (for a new game)' move. Thirdly, knowledge about the speaker's mental state is relevant: if the hearer knows that the speaker should know, or has at least been told, P, then an utterance which looks superficially like a 'query-wh' or 'query-yn' move is probably more likely to be a 'check'. If it is categorised as a check then that hypothesis in turn would weaken the likelihood that the speaker is certain of P: you don't check things you are certain of.

We can illustrate this with an example from the 'Autoroute' domain described in Lewin, Russell, Carter, Browning, Ponting, and Pulman (1993) and Lewin and Pulman (1995). In this domain a person interacts with a system to plan an automobile route between places within the UK. The relevant parameters are start and end of journey, with optional information like type of car, stops on the way, whether to optimise for speed or distance, avoid or follow motorways etc. The system engages in a dialogue to instantiate as many of these parameters as possible and then sends the information to a commercial PC package (described in (NextBase 1991)) which calculates the optimal route.

We can encode the observations described earlier into a network representing the influence of these factors on the recognition of conversational moves. Assume that there are only a finite number of types of proposition $P_1 ... P_n$ which can arise in our domain (which will usually be the case for the kind of simple task-oriented dialogues we are considering, even though there may be an infinite number of ways of expressing them). They will be simplified representations of the propositional content of actual utterances, for example:

destination=cambridge; no; ok; origin=what; etc.

We will assume the variables and values described below (these are just for illustration: in reality, determining the precise form of the network can only be done in conjunction with a close analysis of the corpus dialogues).

Pr: Previous-move = query-yn(P_i),reply-n,....

C: Content and form = positive,negative,ynq(P_i),whq(P_i),dcl(P_i)

K: S-knows-P = yes,no,maybe

M: Current-move = query-yn(P_i),reply-n,....

We also want the results of a particular move classification to feed into an updated model of the speaker's current beliefs. This can be achieved by using the move categorisation network to provide evidence that instantiates a value of a variable in another network. We indicate this in the diagram below by a dotted line connection between node K and an independent subnetwork representing the hearer's beliefs about the speaker's beliefs.

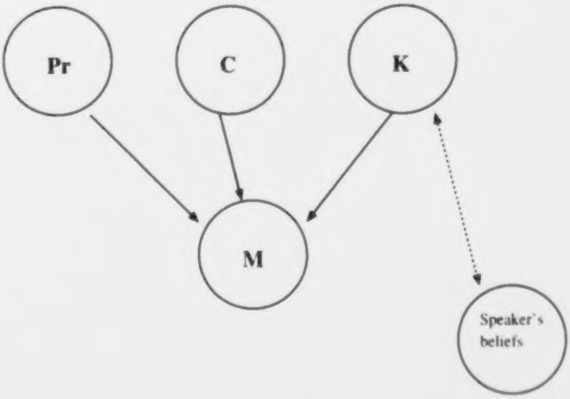


Figure 3: Bayesian net for move recognition

Having decided on the structure of the network, we need to assign *a priori* probabilities to the various values of the variables. This should be done on the basis of statistics derived from annotated corpora, although in many applications estimates of probabilities derived from experts have proved to be quite accurate. In our example, we will assume that the top three nodes may have a fairly uniform initial *a priori* distribution on them, reflecting the fact that in the absence of any evidence, there is no previous move more likely than any other, no proposition more salient than any other, and no hypothesis about the other's beliefs more detailed than any other. However, we can provide some conditional probabilities which express *a priori* dependencies between particular values of M and its parents, e.g.

```
P(M=query-yn(Q)|C=ynq(Q),K=yes,...)          = very low
```

The probability that a yes-no question about Q realises a 'query-yn' move when the other is believed to already know the answer to Q is very low. You don't ask questions about things you already know. (Notice that there is a clear connection here with the notion of preconditions for speech acts. The analogous link between preconditions and moves is reflected in the assignment of probabilities).

```
P(M=query-yn(Q)|C=ynq(Q),K=no,...)           = high
```

A yes-no question is more likely to realise a query move if the speaker is believed not to know the answer already.

```
P(M=check(Q)|C=ynq(Q),K=no,...)                   = low
P(M=check(Q)|C=ynq(Q),K=maybe,...)                = a bit higher
P(M=check(Q)|C=ynq(Q),K=maybe,Pr=reply-wh(R))     = pretty high
... etc.
```

A yes-no question is most likely to be expressing a checking move if the speaker may not know the answer and the previous move was a reply to a question.

```
P(M=ready|Pr=query-yn,C=ok,...) = very low
```

The probability that 'yes', or 'ok' realises a ready move when the previous move was a query is rather low.

```
P(M=ready|Pr=reply-n,C=ok,...) = quite high
```

The probability that 'yes', or 'ok' realises a ready move when the previous move was one which can close a game is quite high.

On the assumption that we have a complete and plausible set of probabilities like this we can give a hypothetical illustration of how such a network might be used in the first few turns of our illustrative sequence.

The basic cycle (from the point of view of one person, here the user) is:

1. instantiate C (and Pr and K - from the speaker's belief network- if possible), update probabilities.

2 find the value of 'move' that maximises $P(M = move \mid Pr, C, K)$

3. instantiate M to 'move', propagate revised probabilities.

4. find value of 'k' that maximises $P(K = k \mid Pr, C, M)$, and feed into speaker's beliefs sub-network.

5. re-initialise main network, and go to 1.

Of course, we also need to provide for the user making their own move, and updating other networks as well.

We can illustrate with our earlier example dialogue:

1. w: Where would you like to go?

We assume for illustration that this is the opening move and so Pr is not instantiated. C is instantiated as 'whq(destination)'. K is not yet instantiated. We will further assume that given the *a priori* probabilities the most likely move for for a wh-question under these circumstances is a wh-query, and so that is the answer at step 2. We now set the value of M to 'wh-query', and propagate the resulting probability changes. Given our estimated conditional probabilities and the new instantiated nodes the value for k that maximises $P(K = k \mid Pr, C = whq, M = wh - query)$ will be 'no', and so the proposition that, at that stage in the dialogue, the speaker does not know the destination is added to the record of beliefs built up in the subnetwork.

The user then plans and executes his own move (exactly how this is done we will return to below):

2. c: Edwinstowe (reply-whq)

Back comes the reply:

3. w: Edwinstowe?

This time round the cycle, Pr='reply-whq', C=ynq(destination=Edwinstowe), and K is 'yes' for the proposition 'destination=Edwinstowe'. This latter value we assume to be a consequence of the user's previous reply: normally you would expect someone to know something they have just been told. This information can be recovered from the 'speaker's belief' subnetwork.

We assume that given the probabilities above, the most likely move assignment for this yes-no question is as a 'check' rather than a genuine question. So we now instantiate M for this value. Recalculating probabilities the most probable value for K with respect to these instantiations should now be 'maybe' rather than 'yes' and this can be used to update the record of speaker beliefs being built up.

5.1 Choosing the next move

Bayesian networks can be extended so as to represent information not only about probabilities, but also utilities attached to the consequences of particular actions. This enables the integration of reasoning about the probability of an effect along with the desirability of that effect. Utilities can be combined and propagated by essentially the same algorithm as is used for probabilities (Neapolitan (1990) esp. Chapter 9).

Bayesian networks extended in this way are usually referred to as 'causal influence diagrams'. To the set of nodes representing propositional variables we add one or more 'decision' nodes, representing a choice about whether or not to perform an action, and exactly one 'value' node, where all the utilities associated with the different actions are represented. If there is more than one decision node, later ones must be dependent on earlier ones.

As an illustration, we will take a network representing the decision whether to start a new game, or to check the previous move. There are consequences associated with these choices, and also there are several factors which we want to influence the choice that is made. The consequences of choosing to check will typically be that the overall dialogue will take longer. However, there is a lesser risk of a misunderstanding or an error causing problems later on. Going on to a new game will typically speed up the dialogue, but if the previous piece of information has not been properly 'grounded' then it may turn out to be insufficient to proceed at some later stage. For example, in our illustration, the wizard might have got the wrong 'Edwinstowe', leading to either an inaccurate route, or a later repeat of most of the dialogue. If speed is important, it might be preferable to move to a new game as soon as possible provided there is reasonable confidence that understanding has been achieved, whereas if accuracy was preferred to speed, frequent checking moves and a more cautious dialogue style would be called for.

The decisions have consequences, but the consequences might also be dependent on other causal factors. For example, if the environment is a noisy one, or the speech recogniser is unreliable, it may be that frequent checking will lead to a better accuracy/speed ratio than a less cautious strategy. Thus a decision will be a calculation based on the likelihood of the effects given the prevailing circumstances, and the utilities associated with those different possible outcomes.

We can illustrate this with the following partial network for deciding which move to make next. In this network we have to choose to check the last move, or start a new game. The causal consequences of this decision are represented by a 'speed' node, saying whether the dialogue is likely to be completed quickly or not, and an 'accurate' node, saying how likely it is that the route given is actually the one asked for.

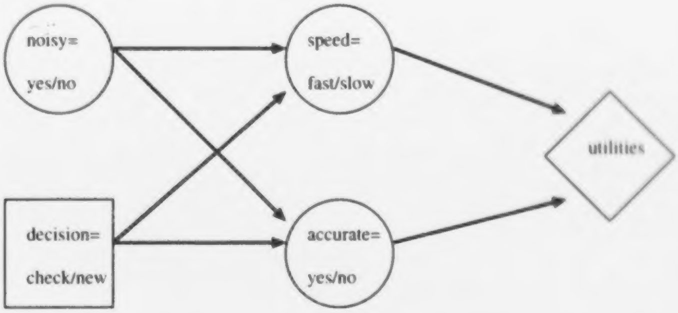


Figure 4: Bayesian Net for Move Choice

The round nodes are propositional variables, as before. (In the 'causal influence diagram' literature they are referred to as 'chance' nodes). The square one is a decision node, representing the choice of possible actions. Although in our example this is not the case, chance nodes can have arcs to decision nodes.

The value node is represented as a diamond. The value node can be regarded as a propositional variable which contains one utility value for each possible combination of its parent nodes. These utilities can be computed from those assigned to the parents, or assigned directly. (The fact that there is only one such node makes the DAG look as if is no longer singly connected. But provided that the subgraph consisting of the chance nodes remains singly connected that does not matter, since the value node does not affect any of the probabilities).

We assign probabilities to chance nodes as before. The probabilities depend both on parent chance nodes, and on whether a particular decision is taken. Thus, for example, the assignment of probabilities to the 'speed' node will depend on what decision was taken, and on how reliable the communication channel is. The precise values we assign to these probabilities do not matter for the sake of the illustration, but we would want the probabilities and the utilities to obey the following constraints:

`P(fast|check,noisy) > P(fast|check,~noisy)`
A check is more likely to lead to an overall speedup in a noisy environment.

`P(accurate|check,noisy) > P(accurate|newgame,noisy)` A check is more likely to maximise accuracy in a noisy environment than moving on to a new game.

`U(fast,accurate) > .... > U(slow,~accurate)` We prefer fast, accurate dialogues. Slow inaccurate ones are of course the worst of all worlds.

Given a network like this with utilities and probabilities assigned, we can calculate for any action its expected utility with respect to the current instantiations of chance nodes. Let a be an action (i.e. a value of the decision node), and let the the current instantiations of chance nodes be represented by e (=evidence). The value node will describe the utility of the causal effects $C_{1 \ldots n}$ of each action, where the notation for such a utility measure is $u(c_i)$:

$$U(a) = \prod_i u(c_i) \times P(c_i \mid a, e)$$

Now we choose the action that has the maximum overall utility under the circumstances, execute the conversational move corresponding to it, and update variables, etc. We would hope that in our example scenario, given the circumstance that the reliability of the communication channel is low, and that the utility that is to be maximised is accuracy, then the decision that would score the highest would be to do a checking move rather than begin a new game.

5.2 Higher level planning

So far we have seen how it is possible to recognise utterances as realising particular conversational moves, and how to select the maximally useful next move, while updating and combining information of several different sorts. Using Bayesian networks, augmented with utility calculations, offers the promise of being able to model locally rational conversational behaviour in a way that has not so far proved possible in practice on a large scale for the traditional BDI-based systems. However, while a system based upon the components we have sketched so far would be a satisfactory 'reactive' system, we have not yet shown how to reproduce the higher levels of strategic planning that are one of the strong points of the traditional architectures.

However, at least as far as relatively simple task-oriented dialogues of the Autoroute, Verbmobil, or ATIS types are concerned, it seems quite possible to extend this scheme to completely replace the traditional types of planning that most dialogue systems rely on for their overall strategy. The analogy here is with the use of decision networks in expert systems, particularly medical diagnosis systems. Here the diagnosis does not necessarily proceed by going through some fixed sequence of questions; rather, the most informative next question is chosen dynamically by testing to see which propositional variable it would be most useful to know the value of. For example, knowing the age of a patient is a very important piece of information, even if not directly relevant to a diagnosis. If the patient is a child, questions about level of alcohol intake are unlikely, even in these times, to yield much diagnostically relevant information. Thus the utility of asking a question about age may be high in terms of speedy diagnosis, even though the answer itself may not be directly relevant.

In the case of our Autoroute domain, we might have variables corresponding to the main parameters of an Autoroute query: start, destination, car type, etc. It is difficult to think of an assignment of utilities that is not rather trivial: for example, we clearly need to know the start and the destination, and so the utilities associated with those variables should be higher than those of e.g. car type. Also, of course, the utility of asking questions about the values of variables that are already instantiated is likely to be very low. However, we might complicate the picture by making the utility of some variables dependent on the values of others: for example, if we know that the user has a fast car, then it is probably less important to ask whether he is interested in a scenic route for his journey. If the user wants to avoid motorways, then he is probably not interested in the fastest as opposed to the shortest journey.

Given a decision network having the general form of those above, and encoding these specific dependencies and utilities, it is possible to decide which propositional variable should be sampled next by the following means.

1. Given a variable with m values: $V_{1 \ldots m}$, then for action a, evidence e, causal effects $C_{1 \ldots n}$ of a, we can calculate the utility of an action with respect to the value of a variable by the following expression:

$$U(a \mid V_i) = \prod_{j=1 \ldots n} u(C_j \mid a, V_i) \times P(C_j \mid a, e, V_i)$$

This expression is related to that used earlier for calculating the utility of a move: the difference is that there, the relevant variable, V, was assumed to be instantiated already.

2. Now we can define the utility for each value of V as:

$$U(V_i) = max_a U(a \mid V_i)$$

This expression tells us the maximum utility that can theoretically be derived from

this value of the variable.
3. Now the overall utility of querying V can be computed by summing the product of the utilities and the likelihood of realising them under the current set of evidential instantiations:

$$\prod_{i=1\ldots m} P(V_i \mid e) \times U(V_i)$$

Performing these computations for all uninstantiated variables will allow the most useful one to be questioned next: the variable that has the highest overall possible utility in the current circumstances is a rational choice for the subject of the next question. Of course, in a large network these calculations might be rather expensive: a practical method might involve some kind of stochastic sampling of variables rather than an exhaustive comparison.

Related Work

The notion of 'language game', 'conversational game' or 'dialogue game' has a long history in 20th century philosophy of language, starting with Wittgenstein. Games interpreted in a decision-theoretic way have also been used within philosophy of language, notably by Hintikka, although within computational linguistics this line of enquiry is probably best known, in one version at least, through the work of Carlson (1983). However, the most direct inspiration for the approach described here is a paper by Gamback, Rayner, and Pell (1991), in which they describe a hybrid rule-based/neural network approach to the pragmatics micro-world of bidding in bridge, in which bids are seen as various kinds of simple speech act.

Bayesian Networks have been used in natural language processing for story understanding (see, for example, Charniak and Goldman (1991)) and word-sense disambiguation. They have also been used by Araki, Kawahara, and Doshita (1995) for dialogue understanding, although in a somewhat different way than envisaged here. The system they describe uses two networks: one 'Conversational Space' network is responsible for hypothesising the interpretation of an utterance and the associated speaker intention. It combines syntactic, semantic, and discourse structure information into a single network, which is constructed dynamically for each new utterance. The second network ('Problem Solving Space') encodes a model of the task domain and is responsible for plan recognition, and for selecting the appropriate type of response. Other mechanisms (e.g. utterance type trigrams) are also used, and 'mental state' is modelled separately, apparently not by a Bayesian network. Explicit utilities and the framework of causal influence diagrams are not used.

Conclusions

We began with three questions that should be answered by any satisfactory computational theory of dialogue. It is worth spelling out the kinds of answers that are given to these questions by the framework we have sketched.

(i) what are mental states? - in the Bayesian network approach, mental states are represented by sets of propositions linked by causal (or logical) relations, with a probability distribution on them that respects these causal relationships. There is a straightforward interpretation of these networks as networks of beliefs, and indeed that is how they were originally envisaged in Pearl (1988). When Bayesian networks are augmented with the apparatus of decision and value nodes, and utilities, then it is plausible to think of them as modelling some aspects of desire and and possibly intention, although the correspondence is not exact. This type of Bayesian reasoning is less powerful than that assumed in classical belief revision or associated frameworks like dynamic logic. Quantificational reasoning, beliefs about beliefs, etc. can only be handled to the extent that they can be 'compiled out' to a propositional format. However, classical BDI implementations have not been able to actually make use of this extra power on a large scale yet and so it remains to be seen whether this is a serious practical constraint.

(ii) how do they change? - states change by the instantiation of nodes representing new evidence, and the consequent updating of probabilities. Conflict between beliefs or intentions in the presence of new input is not modelled explicitly, but can be associated with large differences between *a priori* probabilities and values derived from new evidence. Evidence from multiple sources can be combined unproblematically.

(iii) how do utterances connect with them and change them? - the connection between utterance types and mental states is conventionalised via conversational games, and this conventional connection is encoded in the structure of the relevant networks for move recognition and response. Many of the insights of speech act theory and the BDI tradition are retained and encoded in this way, although their interpretation is now partly probabilistic rather than strictly logical.

Clearly, there is great deal of work to be done before the preceding ideas can be implemented and tested in detail. However, we regard this as a promising perspective from which to approach the problem of building dialogue understanding systems. The Bayesian network architecture seems to provide the right combination of rule based and statistical methods. We can retain what is intuitively correct about the BDI tradition, while overcoming the difficulties and fragilities associated with strictly axiomatic systems.

One obvious question of course, is: where do the networks and their associated probabilities come from? Although it is possible in principle to learn the structure of a Bayesian net from examples, we feel that it is more productive at least in the short term to think of their basic structure as reflecting (corpus-guided) linguistic descriptions of conversational game and move structure. However, the probabilities associated with the nodes in a network should reflect observed properties in a relevant corpus, and it is quite plausible to think of these as being automatically trained from an annotated corpus.

Acknowledgements

Most of the work described here was carried out under a contract to SRI from the Speech Research Unit, DERA Malvern, Contract CSM3/002, and the current paper is a revised and extended version of part of the final report for that project. As well as at CLIN 96 in Eindhoven, versions of the paper have been given to the Speech Research Unit, DERA Malvern; and the Human Communication Research Centre, Edinburgh. I am grateful for the comments received on all these occasions.

I would also like to acknowledge the many valuable inputs to this work by Ian Lewin.

References

Allen, J., B. Miller, E. Ringger, and T. Sikorski (1996). A robust system for natural spoken dialogue. In *Proceedings of 34th ACL Santa Cruz*, pp. 62–70.

Araki, M., T. Kawahara, and S. Doshita (1995). Cooperative spoken dialogue model using bayesian network and event hierarchy. In *Proceedings of ESCA Workshop on Spoken Dialogue Systems, Denmark*, pp. 177–180.

Bunt, H. (1997). Dynamic interpretation and dialogue theory. In M. Taylor, D. Bouwhuis, and F. Neel (Eds.), *The structure of multi-modal dialogue*, Volume 2. Amsterdam John Benjamins.

Carlson, L. (1983). *Dialogue Games*. Dordrecht: D. Reidel Publishing Co.

Charniak, E. and R. Goldman (1991). A probabilistic model of plan recognition. In *Proceedings of Ninth National Conference on AI*, pp. 160–165. AAAI.

Clark, H. and E. Schaefer (1989). Contributing to discourse. *Cognitive Science 13*, 259–294.

Cohen, P. and H. Levesque (1990). Rational interaction as the basis for communication. In P. Cohen, J. Morgan, and M. Pollack (Eds.), *Intentions in Communication*. MIT Press.

Cohen, P., J. Morgan, and M. Pollack (1990). *Intentions in Communication*. MIT Press.

Cohen, P. and C. Perrault (1979). Elements of a plan-based theory of speech acts. *Cognitive Science 3*(3), 177–212.

Galliers, J. (1990). Belief revision and a theory of communication. Technical Report Technical Report 193, University of Cambridge Computer Laboratory.

Gamback, B., M. Rayner, and B. Pell (1991). Pragmatic reasoning in bridge. SRI Cambridge technical report CRC-030, http://www.cam.sri.com/.

Gärdenfors, P. (1988). *Knowledge in flux: modeling the dynamics of epistemic states*. MIT Press.

Groenendijk, J. and M. Stokhof (1991). Dynamic predicate logic. *Linguistics and Philosophy 14*, 39–100.

Hamblin, C. (1971). Mathematical models of discourse. *Theoria 37*, 130–155.

Houghton, G. (1986). *The Production of Language in Dialogue*. Ph. D. thesis, University of Sussex.

Jaspars, J. (1996). A modal unification of dynamic theories. Building the Framework: FraCas Deliverable D15.

Konolige, K. (1986). *A Deduction Model of Belief*. London: Pitman.

Kowtko, J., S. Isard, and G. Doherty (1992). Conversational games within dialogue. HCRC research paper RP-31.

Lewin, I. and S. Pulman (1995). Inference in the resolution of ellipsis. In *Proceedings of ESCA Workshop on Spoken Dialogue Systems, Vigso, Denmark*, pp. 53–56.

Lewin, I., M. Russell, D. Carter, S. Browning, K. Ponting, and S. Pulman (1993). A speech-based route enquiry system built from general-purpose components. In *Eurospeech '93: Proceedings of the 3rd European Conference on speech communication and technology*, Volume 3, pp. 2047–2050.

Moore, R. and S. Browning (1992). Results of an exercise to collect 'genuine' spoken enquiries using woz techniques. In *Proceedings of the Institute of Acoustics 14 6*, pp. 613–620.

Neapolitan, R. (1990). *Probabilistic reasoning in expert systems: theory and algorithms*. Wiley.

NextBase (1991). Autoroute plus user guide. NextBase Limited, Headline House, Chaucer Road, Ashford, Middlesex, England.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufman.

Perrault, C. and J. Allen (1980). A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics 6*(3-4), 167–182.

Power, R. (1979). The organization of purposeful dialogues. *Linguistics 17*, 107–152.

Reithinger, N. and E. Maier (1996). Utilizing statistical dialogue act processing in verbmobil. In *Proceedings of 33rd ACL, Cambridge Mass.*, pp. 116–121.

Sadek, D., A. Ferrieux, A. Cozannet, P. Bretier, F. Panaget, and J. Simonin (1996). Effective human computer cooperative spoken dialogue. In *Proceedings ICSLP 96 Conference in Spoken Language Processing, Philadelphia*, pp. 546–549.

Searle, J. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge Univeristy Press.

Shoham, Y. (1994). *Artificial Intelligence Techniques in Prolog*. San Francico: Morgan Kaufman.

Traum, D. and J. Allen (1994). Discourse obligations in dialogue processing. In *Proceedings of the 32nd ACL, Las Cruces, New Mexico*, pp. 1–8.

Traum, D. and E. Hinkelman (1992). Conversation acts in task-oriented spoken dialogue. *Computational Intelligence 8*(3).

Valence Alternation without Lexical Rules

Gosse Bouma*

Abstract

Valence changing lexical rules are a problematic component of constraint-based grammar formalisms. Lexical rules of this type are procedural, require defaults, and may easily lead to spurious ambiguity. Relational constraints can be used to eliminate such rules. The relational approach does not require defaults, is declarative, avoids spurious ambiguity, and can be an integrated part of a hierarchically structured lexicon. This is illustrated below for the complement extraction and adjunct introduction lexical rules of HPSG. We argue that, apart from the technical benefits mentioned above, the relational approach is linguistically superior, in that it offers a uniform account of complement and adjunct extraction. Furthermore, it eliminates the spurious ambiguity that may arise in grammars which include complement inheritance verbs as well as a lexicalist account of complement extraction.

Introduction

Recent work in HPSG has argued for lexicalist approaches to complement extraction (Sag 1995), adjunct selection (Miller 1992; Iida, Manning, O'Neill, and Sag 1994; Manning, Sag, and Iida 1996), and clitic climbing (Sag and Miller, to appear). Lexicalist accounts treat these phenomena as valence variation. That is, complement extraction requires that each head selecting for an extractable complement C has a counterpart which does not select for C but instead includes C in its SLASH-set. Similarly, lexicalist adjunct selection requires that heads may include (an arbitrary number of) adjuncts on their COMPS-list. Lexicalist accounts of clitic climbing, finally, require not only that lexical heads may realize some of their complements as phonological clitics, but also that these heads have a COMPS-list which is the **append** of the list of elements the head subcategorizes for and the COMPS-list of one of these elements. That is, verbs allowing for clitic climbing must be *complement inheritance* verbs. Note that both the phonological realization of complements as clitics and complement inheritance lead to valence alternations.

The proposals cited above all rely on lexical rules to account for certain systematic alternations in lexical entries. The central role of lexical rules is remarkable, given the fact that lexical rules are often seen as more or less *ad hoc*, procedural, extensions of the formalism, whose formal status is far from resolved.

At the same time, it has been customary in HPSG to employ relational, recursive, constraints. That is, in accounts of phenomena such as extraction, complement inheritance, quantifier scoping, and word order, the values of list and set-valued features are routinely defined using relations such as **member**, **delete**, **append**, and **(sequence) union**. Lexical rules in particular, are often defined using such relations.

---

*Rijksuniversiteit Groningen, vakgroep Alfa-informatica

Finally, the treatment of valence in recent versions of HPSG is more subtle than in the versions presented in Pollard and Sag (1987) and Pollard and Sag (1994, chapters 1-8). Following Borsley (1989), it is now customary to distinguish subjects from other complements by means of the two valence features SUBJ and COMPS (replacing the single SUBCAT feature used before). Furthermore, valence is distinguished from *argument structure*, represented by the feature ARG-ST. Argument structure contains the list of elements which a lexical sign selects for and is the level of representation to which the binding principles apply. While in the canonical case ARG-ST will correspond to the **append** of SUBJ and COMPS, this is by no means always true. In Manning and Sag (1995) it is observed that phenomena such as passive, 'pro-drop', and syntactic ergativity in a number of languages can be seen as evidence for several non-canonical relationships between valence and argument structure, providing evidence for a level of representation independent of valence. Note also that complement inheritance verbs will typically contain (inherited) elements on COMPS that do not correspond to arguments of that verb. Lexicalized extraction, finally, implies that some (non-subject) elements on ARG-ST will not be present on COMPS, but are included in SLASH instead.

In this paper, it is argued that the distinction between valence and argument structure allows valence changing lexical rules to be eliminated. Valence alternations are captured instead by general, possibly recursive, constraints defining the mapping between argument structure and valence. We demonstrate this in some detail for complement extraction and adjunct introduction. Other valence changing lexical rules (such as) can in principle be replaced by constraints in a similar fashion.[1]

Approaches to valence variation using relational constraints have been proposed by, among others, Kathol (1994) and Frank (1994). The current proposal, however, allows recursive constraints, and thus it can account for complement extraction (requiring arbitrary elements on ARG-ST to be realized as *gaps*) and adjunct introduction (requiring the insertion of an arbitrary number of adjuncts on ARG-ST (and COMPS)). Furthermore, the constraints proposed below do not require the introduction of additional features. Instead, all constraints apply to independently motivated features, leading to a tight integration of the constraint system with the overall architecture of HPSG.

The connection between lexical rules and relational constraints was first noted in van Noord and Bouma (1994). By viewing lexical rules as relational constraints, delayed evaluation techniques can be used to solve the computational problems posed by recursive lexical rules. However, the constraints in van Noord and Bouma (1994) hold between full-blown lexical entries (i.e. signs), whereas below we use constraints to relate only specific features within a sign. Since all constraints apply to the same sign conjunctively, the issue of rule-ordering, which was solved in van Noord and Bouma (1994) by hard-wiring the order of rule application into the constraints (see Meurers and Minnen (1995) for an alternative approach), disappears. Also, the need for default sharing of information between input and output of a lexical rule disappears.

---

[1] In Sag and Miller (to appear), for instance, an account of French clitization is presented which is directly compatible with (and inspired by) the approach outlined below in that it defines the realization of certain elements on ARG-ST as clitics by means of a constraint on the mapping between ARG-ST and COMPS, instead of by means of a lexical rule, as in previous proposals.

Below, we present an example lexicon fragment, in which both lexical inheritance and lexical rules are used. We point out a number of problematic aspects of these rules in a constraint-based setting. In section 2, we redefine the fragment by replacing lexical rules with relational constraints. We demonstrate that the constraint-based fragment naturally leads to an account of complement extraction which subsumes the possibility of adjunct extraction. In section 3, we argue that the kind of spurious ambiguity noted in Hinrichs and Nakazawa (1996) does not arise in our proposal.

1 A lexicon fragment with lexical rules

We present a lexicon fragment for verbs which uses inheritance, constraints, and lexical rules. We point out various problematic aspects of this set-up.

The basic lexicon

A definite clause specification for the basic lexical entries of a small lexicon fragment is given in fig. 1. The unary predicate `basic-entry` defines the set of basic lexical entries in the language. A basic entry is of type *word*, and can be a major category (i.e. *v*, *n*, *etc.*) entry satisfying the `slash-amalgamation` constraint (introduced below). A verbal major category must satisfy `verbal-subcat` and `map-args`. The first defines the various verbal subcategorization types, whereas the latter defines the mapping between argument structure and valence.

Following Manning and Sag (1995), we assume that different verbal subcategorization types differ only in their argument structure, and that the values of the valence features are defined by means of a general *mapping* constraint. This is the task of the relational constraint `map-args`. Canonically, the first element on ARG-ST is the subject, while the rest is equal to COMPS ('|' connects the head and tail of a list). (Alternative definitions are considered below.). By combining the definitions of `verbal-subcat`, `verbal-lex`, and `map-args`, we can for instance derive the following fact:

$$\text{(1) }\mathtt{major(}\begin{bmatrix} \text{PHON} & \textit{hates} \\ \text{HEAD} & v \\ \text{ARG-ST} & \langle \boxed{1}\ \text{NP}_i,\ \boxed{2}\ \text{NP}_j \rangle \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{2} \rangle \\ \text{CONT} & \textit{hate}'(i,j) \end{bmatrix}\mathtt{)}$$

$$\texttt{basic-entry}(\boxed{1}\begin{bmatrix} \textit{word} & \\ \text{ARG-ST} & \boxed{2} \\ \text{SLASH} & \boxed{3} \end{bmatrix}) \leftarrow$$

$$\qquad \texttt{major}(\boxed{1}) \wedge \texttt{slash-amalgamation}(\boxed{2}, \boxed{3})$$

$$\texttt{major}(\boxed{1}\begin{bmatrix} \text{HEAD} & v \\ \text{ARG-ST} & \boxed{2} \\ \text{SUBJ} & \boxed{3} \\ \text{COMPS} & \boxed{4} \end{bmatrix}) \leftarrow$$

$$\qquad \texttt{verbal-subcat}(\boxed{1}) \wedge \texttt{map-args}(\boxed{2}, \boxed{3}, \boxed{4})$$

$$\texttt{verbal-subcat}(\begin{bmatrix} \text{PHON} & \boxed{1} \\ \text{ARG-ST} & \langle\ \text{NP}_i\ \rangle \\ \text{CONT} & \boxed{2}(i) \end{bmatrix}) \leftarrow$$

$$\qquad \texttt{verbal-lex(intrans,}\boxed{1}\texttt{,}\boxed{2}\texttt{)}$$

$$\texttt{verbal-subcat}(\begin{bmatrix} \text{PHON} & \boxed{1} \\ \text{ARG-ST} & \langle\ \text{NP}_i, \text{NP}_j\ \rangle \\ \text{CONT} & \boxed{2}(i, j) \end{bmatrix}) \leftarrow$$

$$\qquad \texttt{verbal-lex(trans,}\boxed{1}\texttt{,}\boxed{2}\texttt{)}$$

$$\texttt{verbal-lex(instrans,}\textit{sleeps}\texttt{,}\textit{sleep}'\texttt{)}$$

$$\texttt{verbal-lex(trans,}\textit{hates}\texttt{,}\textit{hate}'\texttt{)}$$

$$\texttt{map-args}(\langle\ \boxed{1} \mid \boxed{2}\ \rangle, \langle\ \boxed{1}\ \rangle, \boxed{2}\ )$$

$$\texttt{slash-amalgamation}(\langle\ \rangle, \emptyset)$$

$$\texttt{slash-amalgamation}(\langle\ [\ \text{SLASH}\ \boxed{1}\ ] \mid \boxed{2}\ \rangle, \boxed{1} \uplus \boxed{3}) \leftarrow$$

$$\qquad \texttt{slash-amalgamation}(\boxed{2}, \boxed{3})$$

Figure 1: A fragment of the basic lexicon

The two main features of the lexicalist approach to extraction presented in Sag (1995) is the elimination of the NONLOCAL FEATURE PRINCIPLE in favour of a lexical *slash amalgamation* constraint and the elimination of traces in favour of a lexical complement extraction rule. Slash amalgamation requires that the SLASH-value of a basic lexical entry is the set-union of the SLASH-values of its arguments. The **slash-amalagamation** constraint implements this by recursively traversing the list of elements on ARG-ST, and unioning all the SLASH values: (⊎ denotes (non-vacuous) set-union). Slash amalgamation makes the NONLOCAL FEATURE PRINCIPLE superfluous, as SLASH can simply be shared between head and mother in phrases without a filler daughter, while SLASH is subject to rule-specific constraints in *head-filler* phrases. An example of slash amalgamation at work is given after we have introduced the lexical rule for extraction.

The basic lexicon incorporates the following notion of *lexical inheritance*: A basic entry has various major category entries as its subclasses. All of these subclasses must satisfy **slash-amalgamation**. Similarly, the verbal major category class has various verbal subcategorization types as subclass. All of these must satisfy **map-args**. Thus, the unary predicates in general define subclasses of the general class **basic-entry**, whereas the other predicates define constraints which must hold for the class in whose antecedent the predicate appears.

Adding lexical rules

In fig. 2, we define two lexical rules. A lexical rule defines a relationship between an 'input' and 'output' lexical entry. Therefore, lexical rules can be added to the fragment as instances of the relation **lexical-rule**(*In*,*Out*). Furthermore, the set of lexical entries (basic or derived) is now defined by the relation **entry**.

$$\texttt{entry}(\boxed{1}) \leftarrow$$

$$\qquad \texttt{basic-entry}(\boxed{1})$$

$$\texttt{entry}(\boxed{1}) \leftarrow$$

$$\qquad \texttt{entry}(\boxed{0}) \wedge \texttt{lexical-rule}(\boxed{0},\boxed{1})$$

%% complement extraction lexical rule (celr)

$$\texttt{lexical-rule}(\begin{bmatrix} \text{COMPS} & \boxed{1} \bigcirc \left\langle \begin{bmatrix} \text{LOC} & \boxed{2} \\ \text{SLASH} & \{\ \boxed{2}\ \} \end{bmatrix} \right\rangle \\ \\ \end{bmatrix}, \begin{bmatrix} \text{COMPS} & \boxed{1} \end{bmatrix})$$

%% adjuncts lexical rule

$$\texttt{lexical-rule}(\begin{bmatrix} \text{ARG-ST} & \boxed{1} \\ \text{CONT} & \boxed{2} \end{bmatrix}, \begin{bmatrix} \text{ARG-ST} & \boxed{1} \oplus \langle\ \text{ADV}\ \rangle \\ \text{CONT} & adv'(\ \boxed{2}\ ) \end{bmatrix})$$

Figure 2: Adding lexical rules

The complement extraction lexical rule (CELR) is adopted from Sag (1995), and identifies an element on COMPS as a *gap* (i.e. subtype of *synsem* which satisfies the constraint that its SLASH-value is a singleton set, the only element of which is reentrant with LOC). The *gap* is absent in the output of the rule (○ denotes sequence union). The interaction of this rule with slash amalgamation implies that the SLASH-value of the deleted element will be included in SLASH of the input (and output) sign. As each complement is also present on ARG-ST, and the SLASH-value of the sign itself is the union of the SLASH-value of its members, the instantiation of SLASH on one of these members will have a direct effect on SLASH. Furthermore, as it is assumed that information is shared by default between input and output, the instantiated SLASH value will be present on the output of the rule as well. Note, however, that the present formulation does not account for default sharing of information.

Assuming the latter problem can be solved, the CELR allows the derivation of the following lexical entry, in which the object has been extracted:

$$(2)\ \texttt{entry}(\begin{bmatrix} \text{PHON} & \textit{hates} \\ \text{ARG-ST} & \left\langle \boxed{1}\ \text{NP}_i [\text{SLASH}\ \boxed{3}],\ \begin{bmatrix} \text{LOC} & \boxed{2}\ \text{NP}_j \\ \text{SLASH} & \{\ \boxed{2}\ \} \end{bmatrix} \right\rangle \\ \text{SUBJ} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{SLASH} & \boxed{3} \uplus \{\ \boxed{2}\ \} \end{bmatrix})$$

Together with slash amalgamation, and the assumption that SLASH is a head feature in head-valence phrases, while it gets 'bound' in head-filler phrases, this allows for the derivations of the example in fig. 3.

The second lexical rule lexically introduces adjuncts as complements. Several versions of such a rule have been presented (Miller 1992; van Noord and Bouma 1994; Manning, Sag, and Iida 1996). Here, we will assume, following Manning, Sag, and Iida (1996), that adjuncts are added to ARG-ST for reasons of binding and (adjunct) extraction (⊕ denotes append).

Again, this rule as given is incomplete. However, an appeal to default matching cannot give the correct results in this case. Note that the newly introduced adjunct should be added to COMPS as well. This means that the value of COMPS on input and output must differ, in spite of the fact that the rule does not mention them. Intuitively, the correct value for COMPS should follow from the **`map-args`** constraint. It is unclear, however, how that constraint could be made to apply at this point. For one thing, the interaction with complement extraction (which creates exceptions to the canonical mapping relation) appears to be highly problematic. That is, a lexical entry derived by means of complement extraction contains an element on ARG-ST which is not realized on COMPS (i.e. the derived entry for *kiss* above). Adding an adjunct to such an entry and reapplying **`map-args`** to the result would reintroduce the extracted complement.

A similar difficulty arises in trying to account for adjunct extraction. The CELR relies on the fact that **`slash-amalgamation`** takes into account all elements on ARG-ST. However, the adjuncts rule introduces new elements on ARG-ST. This

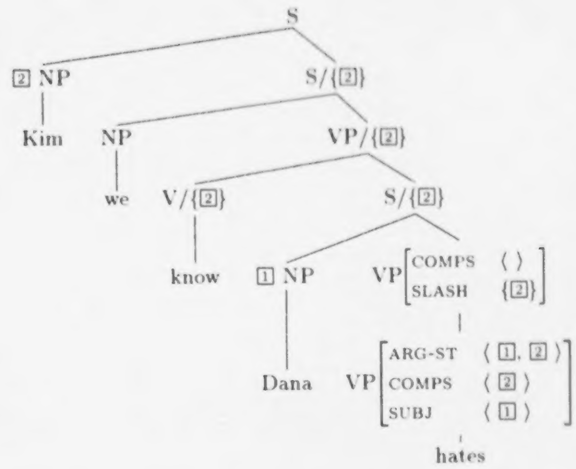

S
2 NP
S/{2}
Kim
NP
VP/{2}
we
V/{2}
S/{2}
know
1 NP
VP [COMPS ⟨ ⟩, SLASH {2}]
Dana
VP [ARG-ST ⟨ 1, 2 ⟩, COMPS ⟨ 2 ⟩, SUBJ ⟨ 1 ⟩]
hates

Figure 3: Kim, we know Dana hates

implies that extracting an adjunct by means of the CELR only has the intended effect if `slash-amalgamation` is used to 'recompute' the value of SLASH on the output of the rule.

Problems for lexical rules

We conclude this section with an overview of problematic aspects of lexical rules in a constraint-based setting.

Default sharing between input and output. Lexical rules typically affect only a small part of the information in a lexical entry. To account for the similarity between input and output, some kind of default sharing of information is required. Default unification as defined in Bouma (1992), Carpenter (1992) or Lascarides, Briscoe, Asher, and Copestake (1996) either is not applicable to the typed constraint language presupposed by HPSG or to the problem of default sharing in lexical rules. Therefore, Meurers (1995) proposes a special-purpose default mechanism for lexical rules. Even if this problem can be solved, it is still the case that lexical rules are the only component of HPSG where nonmonotonicity comes into play.

Interaction with Inheritance. The adjuncts lexical rule illustrates clearly that in some cases one wants to use inheritance of constraints to fill in missing information in the output, instead of default sharing. The discussion of lexical rules in Pollard and Sag (1987, chapter 8) also makes this assumption. No detailed proposals for such an interpretation of lexical rules exist, however. The conflict between these two interpretations of lexical rules also seems to have gone unnoticed in the literature.

Spurious ambiguity. The complement extraction lexical rule removes an element from COMPS. If this rule is used to delete two elements, say C_1 and C_2, one could either remove C_1 first, or C_2. The distinction is irrelevant for the outcome, however. Similarly, complement extraction removes an element, while adjunct introduction adds an element. Again, both orders are possible, but in general (the exception being cases of adjunct extraction) this will lead to the same result. To eliminate this kind of redundancy, one may have to introduce external rule ordering, reformulate the rules so that they no longer need to apply recursively (van Noord and Bouma 1994), or add finite state control devices (Meurers and Minnen 1995).

Subsumption. Hinrichs and Nakazawa (1996) have argued that lexical rules should only be applied to lexical entries that are subsumed by the input conditions of the rule. However, not all lexical rules can be interpreted this way. Also, checking for subsumption appears to be incompatible with certain processing strategies. We return to this issue in section 3.

2 A constraint-based alternative

A radical solution for the problems just mentioned is to eliminate lexical rules and to account for valence variation by means of (recursive) constraints only. On the one hand, the elimination of lexical rules is a substantial simplification of the formalism. On the other hand, using recursive constraints for valence alternations is not a complication of the formalism, as recursive constraints are used in various other components of HPSG already. Note also that lexical rules in particular tend to be defined in terms of recursive constraints. That is, arguments that a system with lexical rules allows fewer or simpler constraints than a system without lexical rules can be rejected easily.

The fragment presented in section 1 defines the relationship between argument structure and valence by means of a mapping constraint. Valence changing lexical rules, such as the complement extraction lexical rule, typically derive lexical entries which do not obey the 'canonical mapping'. A more principled approach to valence alternations, therefore, is to take these lexical entries not as exceptions, derived by means of a rule, but to redefine the mapping between argument structure and valence, so as to allow for the 'exceptional' cases as well.

The definitions in fig. 4 provide a reformulation of the definition of major verbal category lexical entries presented in fig. 1. The CELR and adjuncts lexical rules are made superfluous by a reformulation of `map-args` and the introduction of an `adjuncts` constraint on verbal lexical entries.

Complement extraction

The `map-args` constraint relates argument structure to the valence features SUBJ and COMPS, as before. The new `map-non-subj-args` constraint ensures that non-subject arguments are either realized as complements, or as *gaps*. The range of lexical entries satisfying `map-args` as defined in fig. 4 therefore corresponds exactly to what can be derived by means of the CELR, making the latter spurious.

$$\texttt{major(}\begin{bmatrix}\text{PHON} & \boxed{0}\\ \text{HEAD} & v\\ \text{ARG-ST} & \boxed{1}\oplus\boxed{5}\\ \text{SUBJ} & \boxed{2}\\ \text{COMPS} & \boxed{3}\\ \text{CONT} & \boxed{4}\end{bmatrix}\texttt{)}\leftarrow$$

$$\qquad\texttt{verbal-subcat(}\begin{bmatrix}\text{PHON} & \boxed{0}\\ \text{ARG-ST} & \boxed{1}\\ \text{CONT} & \boxed{6}\end{bmatrix}\texttt{)}\wedge$$

$$\qquad\texttt{adjuncts(}\boxed{5},\boxed{6},\boxed{4}\texttt{)}\wedge$$

$$\qquad\texttt{map-args(}\boxed{1}\oplus\boxed{5},\boxed{2},\boxed{3}\texttt{)}$$

%% map-args(Arg-st,Subj,Comps)

$$\texttt{map-args(}\langle\boxed{1}|\boxed{2}\rangle,\langle\boxed{1}\rangle,\boxed{3}\texttt{)}\leftarrow$$

$$\qquad\texttt{map-non-subj-args(}\boxed{2},\boxed{3}\texttt{)}$$

%% map-non-subj-args(Arg-st,Comps)

$$\texttt{map-non-subj-args(}\langle\ \rangle,\langle\ \rangle\texttt{)}$$

$$\texttt{map-non-subj-args(}\langle\boxed{1}\mid\boxed{2}\rangle,\langle\boxed{1}\mid\boxed{3}\rangle\texttt{)}\leftarrow$$

$$\qquad\texttt{map-non-subj-args(}\boxed{2},\boxed{3}\texttt{)}$$

$$\texttt{map-non-subj-args(}\left\langle\begin{bmatrix}\text{LOC} & \boxed{1}\\ \text{SLASH} & \{\boxed{1}\}\end{bmatrix}\mid\boxed{2}\right\rangle,\boxed{3}\texttt{)}\leftarrow$$

$$\qquad\texttt{map-non-subj-args(}\boxed{2},\boxed{3}\texttt{)}$$

%% adjuncts(Arg-st, Cont, Cont)

$$\texttt{adjuncts(}\langle\ \rangle,\boxed{1},\boxed{1}\texttt{)}$$

$$\texttt{adjuncts(}\langle\text{ADV}\mid\boxed{1}\rangle,\boxed{2},\boxed{3}\texttt{)}\leftarrow$$

$$\qquad\texttt{adjuncts(}\boxed{1},\mathit{adv}'(\boxed{2}),\boxed{3}\texttt{)}$$

Figure 4: A fragment without lexical rules

For instance, assume that the following can be derived by resolving **`major`** with **`verbal-subcat`** and **`adjuncts`**:

$$\text{(3)}\ \texttt{major}\left(\begin{bmatrix}\text{PHON} & \textit{hates}\\ \text{ARG-ST} & \boxed{1}\,\langle\, \text{NP}_i,\ \text{NP}_j \,\rangle\\ \text{SUBJ} & \boxed{2}\\ \text{COMPS} & \boxed{3}\end{bmatrix}\right) \leftarrow \texttt{map-args}(\boxed{1}, \boxed{2}, \boxed{3})$$

This can be resolved with **`map-args`** to give rise to exactly the following two results:

$$\text{(4)}\ \ \text{a.}\ \texttt{major}\left(\begin{bmatrix}\text{PHON} & \textit{hates}\\ \text{ARG-ST} & \langle\, \boxed{1}\,\text{NP},\ \boxed{2}\,\text{NP} \,\rangle\\ \text{SUBJ} & \langle\, \boxed{1} \,\rangle\\ \text{COMPS} & \langle\, \boxed{2} \,\rangle\end{bmatrix}\right)$$

$$\text{b.}\ \texttt{major}\left(\begin{bmatrix}\text{PHON} & \textit{hates}\\ \text{ARG-ST} & \left\langle \boxed{1}\,\text{NP},\ \begin{bmatrix}\text{LOC} & \boxed{2}\,\text{NP}\\ \text{SLASH} & \{\, \boxed{2} \,\}\end{bmatrix} \right\rangle\\ \text{SUBJ} & \langle\, \boxed{1} \,\rangle\\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\right)$$

Note that **`map-args`** imposes a *constraint on* lexical entries, and does not define a *relation between* lexical entries. Since all lexical entries, or at least all verbs, must satisfy **`map-args`**, and since there is no distinction between a 'basic' and a 'derived' lexical entry, the issue of default sharing of information disappears.

A second advantage is that **`map-non-subj-args`** recursively traverses ARG-ST and (nondeterministically) 'decides' for each element whether it is to be realized as complement or as gap. Consequently, the spurious ambiguity that was observed for the CELR in case multiple complements had to be extracted, does not arise in the constraint-based approach.

Adding adjuncts

In section 1, we assumed that the argument structure of a verb is fully determined by its subcategorization type. The adjuncts lexical rule, however, is incompatible with this assumption, as it appends elements to ARG-ST which the verb does not subcategorize for.

The conflict can be resolved by assuming that argument structure is the **append** of two lists ($\boxed{1} \oplus \boxed{5}$ in the definition of **verb** in fig. 4), where the value of the first is determined by the verbal subcategorization type, and the value of the second is determined by the **`adjuncts`** constraint. A similar modification is necessary to account for semantics (CONT). As adjunct semantics takes the basic verbal semantics as argument, the semantics of a verb is no longer directly determined by choosing a particular instance of **`verbal-subcat`**. Instead, **`verbal-subcat`** supplies a basic semantic value which is taken as argument in the `adjuncts` constraint. The latter actually determines the CONT value of `verb`.

The effect of these modifications can be illustrated as follows. Assume that `major` resolves with `verbal-subcat` to give rise to the following:

$$(5)\ \texttt{major}\left(\begin{bmatrix} \text{PHON} & \textit{hates} \\ \text{ARG-ST} & \boxed{1}\ \langle\ \text{NP}_i,\ \text{NP}_j\ \rangle \oplus \boxed{5} \\ \text{SUBJ} & \boxed{2} \\ \text{COMPS} & \boxed{3} \\ \text{CONT} & \boxed{4} \end{bmatrix}\right) \leftarrow$$

$$\texttt{adjuncts}(\boxed{5}, \textit{hate}'(i,j), \boxed{4}) \wedge \texttt{map-args}(\boxed{1} \oplus \boxed{5},\ \boxed{2},\ \boxed{3})$$

Resolution with `adjuncts`, among others, gives:

$$(6)\ \texttt{major}\left(\begin{bmatrix} \text{PHON} & \textit{hates} \\ \text{ARG-ST} & \boxed{1}\ \langle\ \text{NP}_i,\ \text{NP}_j,\ \text{ADV}\ \rangle \\ \text{SUBJ} & \boxed{2} \\ \text{COMPS} & \boxed{3} \\ \text{CONT} & \textit{adv}'(\textit{hate}'(i,j)) \end{bmatrix}\right) \leftarrow \texttt{map-args}(\boxed{1},\ \boxed{2},\ \boxed{3})$$

which in turn can be resolved against `map-args` to give:

$$(7)\ \texttt{major}\left(\begin{bmatrix} \text{PHON} & \textit{hates} \\ \text{ARG-ST} & \langle\ \boxed{1}\ \text{NP}_i,\ \boxed{2}\ \text{NP}_j,\ \boxed{3}\ \text{ADV}\ \rangle \\ \text{SUBJ} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \boxed{2},\ \boxed{3}\ \rangle \\ \text{CONT} & \textit{adv}'(\textit{hate}'(i,j)) \end{bmatrix}\right)$$

The newly introduced `adjuncts` constraint has exactly the same effect as the corresponding lexical rule. Since the constraint is integrated with the lexical hierarchy, however, the mapping between argument structure and valence is automatically accounted for.

Adjunct extraction

Since the possibility of adjuncts on ARG-ST is now taken into account in the definition of verbal lexical entries (i.e. the definition of `major` as given in fig. 4), `slash-amalgamation` will automatically apply to adjuncts on ARG-ST as well. Furthermore, the mapping between argument structure and valence, defined by `map-args`, will also take adjuncts into account. As `slash-amalgamation` and `map-args` are the two constraints responsible for complement extraction, the possibility of adjunct extraction is now just a special case of complement extraction.

For instance, an alternative solution for (6) is:

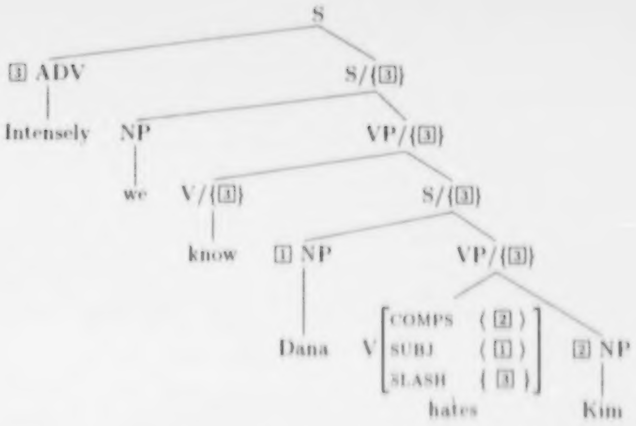


Figure 5: Intensely, we know Dana hates Kim

$$\text{(8)}\ \ \texttt{verb}\left(\begin{bmatrix} \text{PHON} & \textit{hates} \\ \text{ARG-ST} & \left\langle \boxed{1}\ \text{NP}_i,\ \boxed{2}\ \text{NP}_j,\ \begin{bmatrix} \text{LOC} & \boxed{3}\ \text{ADV} \\ \text{SLASH} & \{\ \boxed{3}\ \} \end{bmatrix} \right\rangle \\ \text{SUBJ} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \boxed{2}, \boxed{3}\ \rangle \\ \text{CONT} & \mathit{adv'}(\mathit{hate'}(i,j)) \end{bmatrix}\right)$$

This allows us to derive the entry in (9) for *hates*, where **`slash-amalgamation`** has applied. An example involving this entry is given in fig. 5.

$$\text{(9)}\ \ \texttt{entry}\left({\begin{bmatrix} \text{PHON} & \textit{hates} \\ \text{ARG-ST} & \boxed{0}\left\langle \boxed{1}\ \text{NP}_i[\text{SL}\ \boxed{4}],\ \boxed{2}\ \text{NP}_j[\text{SL}\ \boxed{5}],\ \begin{bmatrix} \text{LOC} & \boxed{3}\ \text{ADV} \\ \text{SL} & \{\ \boxed{3}\ \} \end{bmatrix} \right\rangle \\ \text{SUBJ} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \boxed{2}\ \rangle \\ \text{CONT} & \mathit{adv'}(\mathit{hate'}(i,j)) \\ \text{SLASH} & \boxed{4} \uplus \boxed{5} \uplus \{\ \boxed{3}\ \} \end{bmatrix}}_{\mathit{word}}\right)$$

Constraints declaratively and monotonically define the space of possible lexical entries, whereas lexical entries do this procedurally and nonmonotonically. Therefore, constraints can be integrated into a hierarchical lexicon definition in a way that is difficult or impossible for a system using lexical rules. Furthermore, since the system is declarative, procedural issues such as rule ordering and spurious ambiguity do not arise. Since constraints relate specific features, and not (complete) lexical entries, default sharing of information also is no longer necessary.

There are also linguistic benefits. A grammar avoiding spurious ambiguity is linguistically preferable over a system which does allow spurious derivations. Also, as shown above, the constraint-based approach can account for the possibility of adjunct extraction in a way that does not require any additional rules or mechanisms.

3 Complement Inheritance and Extraction

In this section, we argue that the constraint-based approach also offers a solution for the spurious ambiguity problem observed in Hinrichs and Nakazawa (1996).

Hinrichs and Nakazawa (1994) have argued that German modal and auxiliary verbs are complement inheritance verbs, i.e. they subcategorize for a possibly unsaturated lexical verbal complement, and include the complements of this verb in their own COMPS list. That is, a modal verb such as German *können* (*to be able to*) must be associated with the feature structure in (10). In Hinrichs and Nakazawa (1996) it is argued that a combination of complement inheritance and an approach to complement extraction based on lexical rules leads to spurious ambiguity in sentences containing modal or auxiliary verbs, as an inherited complement may be extracted not only from the COMPS-list of the verb which subcategorizes for it, but also from the COMPS-list of each of the verbs inheriting this complement. This is illustrated in (11).

$$(10)\quad \begin{bmatrix} \textsc{phon} & \textit{können} \\ \textsc{arg-st} & \langle \boxed{1}\, \mathrm{NP}_i,\ \boxed{2}\, \mathrm{V}_j[\textsc{comps}\ \boxed{3}\,]\ \rangle \\ \textsc{subj} & \langle \boxed{1} \rangle \\ \textsc{comps} & \boxed{3} \oplus \langle \boxed{2} \rangle \\ \textsc{cont} & \textit{be-able}'(i, j) \end{bmatrix}$$

(11) $\boxed{1}$ NP $\boxed{2}$ $\mathrm{V}\big[\textsc{comps}\ \langle \boxed{1} \rangle\big]$ $\mathrm{V}\big[\textsc{comps}\ \langle \boxed{1}, \boxed{2} \rangle\big]$

Welches Buch wird Peter kaufen können
`which book will Peter buy be-able`
Which book will Peter be able to buy

The extracted element *welches Buch* appears on the COMPS list of two verbs, and thus a complement extraction lexical rule could apply to either *kaufen* or *können*.

The solution proposed by Hinrichs and Nakazawa (1996) is to let lexical rules apply only to inputs which are subsumed by the input conditions of the rule. Since inherited complements are not instantiated (lexically) on COMPS of a complement inheritance verb, the complement extraction lexical can no longer extract inherited complements. This solution is not without problems, however. First, more recent versions of the CELR, such as the one proposed in Sag (1995), both instantiate and delete an element in the input. Thus, for such a rule it is crucial that the input is not necessarily subsumed by the input conditions of the rule. Second, subsumption appears to be thoroughly incompatible with processing strategies involving delayed evaluation (van Noord and Bouma 1994), a technique which is relevant especially for the type of grammar considered in Hinrichs and Nakazawa (1996). For a subsumption test, the moment of evaluation, and thus the order in which constraints are evaluated, is essential. For delayed evaluation, however, it must be the case that order in which constraints are evaluated can be determined dynamically.

The constraint-based analysis of complement extraction developed in section 2 integrates the account of extraction with the mapping between argument structure and valence. Remember that **map-non-subj-args** determines for each (non-subject) element on ARG-ST whether it is to be realized as a complement or as a *gap*. Consequently, only *arguments* of a verb can be extracted. Since the extracted NP in examples such as (11) above appears on the ARG-ST of *kaufen* only, no spurious ambiguity will arise. Thus, the elimination of lexical rules also eliminates the problem observed in Hinrichs and Nakazawa (1996), without requiring a subsumption test.

The introduction of complement inheritance does present another kind of challenge, however. In the constraint-based fragment presented above, verbal subcategorization types only specify argument structure. The mapping between argument structure and valence is determined by a general **map-args** constraint. Complement inheritance verbs are characterized by the fact that their COMPS-list may contain (inherited) complements which do not correspond to elements of ARG-ST. Consequently, complement inheritance verbs do not obey the **map-args** constraint as defined in the previous section.

Complement inheritance can be accounted for if a rather different characterization of complement inheritance is introduced. Together with a modification of the **map-args** constraint this will make it possible to include complement inheritance verbs in the constraint based fragment developed so far.

Whereas complements are normally saturated phrases (i.e. their COMPS-value is the empty list), the verbal complements of complement inheritance verbs need not be saturated. Thus, in terms of the verbal subcategorization relation introduced in the previous section, the distinction between a regular VP-complement taking verb, such as *versuchen* (*try*) and *können* is that the former requires a saturated VP whereas the latter selects for a (lexical) verbal complement, but does not impose any conditions on the value of COMPS of that complement. The relevant entries for **verbal-subcat** are given below.

(12) $$\texttt{verbal-subcat}(\begin{bmatrix} \text{PHON} & \textit{versuchen} \\ \text{ARG-ST} & \langle \text{NP}_i, \text{V}_j[\text{COMPS}\ \langle\ \rangle] \rangle \\ \text{CONT} & \textit{versuchen}'(i,j) \end{bmatrix})$$

$$\texttt{verbal-subcat}(\begin{bmatrix} \text{PHON} & \textit{können} \\ \text{ARG-ST} & \langle \text{NP}_i, \text{V}_j[\text{COMPS}\ \boxed{1}\ ] \rangle \\ \text{CONT} & \textit{können}'(i,j) \end{bmatrix})$$

Note that $\boxed{1}$ in the second clause is provided only to make the contrast with the first clause explicit. As it is an anonymous variable, no constraint whatsoever is imposed on the value of COMPS. This suffices as a characterization of complement inheritance, if **map-non-subj-args** is modified as follows:

(13)
$$\texttt{map-non-subj-args}(\langle\,\rangle)$$
$$\texttt{map-non-subj-args}(\langle \boxed{1}\,[\textsc{comps}\ \boxed{2}] \mid \boxed{3}\rangle, (\boxed{2} \oplus \langle \boxed{1}\rangle \oplus \boxed{4})\,) \leftarrow$$
$$\texttt{map-non-subj-args}(\boxed{3}, \boxed{4})$$
$$\texttt{map-non-subj-args}(\left\langle \begin{bmatrix}\text{LOC} & \boxed{1}\\ \text{SLASH} & \{\,\boxed{1}\,\}\end{bmatrix} \mid \boxed{2}\right\rangle, \boxed{3}) \leftarrow$$
$$\texttt{map-non-subj-args}(\boxed{2}, \boxed{3})$$

The second clause, which maps non-subject arguments onto COMPS also prepends the complements of this element. This clause applies generally (i.e. to all complements) and thus the possibility of complement inheritance is the rule, rather than the exception. Note, however, that for verbs selecting saturated complements, $\boxed{2}$ in the definition above will be the empty list. In those cases, the definition of **map-non-subj-args** simply works as before. In cases where the value of COMPS of a complement is left unspecified (i.e. the verbal complement of an inheritance verb) the definition has the effect of prepending the complements of the verbal complement on COMPS, and thus a lexical entry will result which is identical to what is proposed in Hinrichs and Nakazawa (1994).

4 Conclusions

We have argued that recursive constraints can be used to eliminate a highly problematic class of lexical rules, i.e. those affecting valence. Apart from avoiding a number of technical difficulties associated with the use of lexical rules, the constraint-based alternative has the advantage of providing a uniform account of complement and adjunct selection without spurious ambiguity.

References

Borsley, R. D. (1989). An HPSG approach to Welsh. *Journal of Linguistics 25*, 333–354.

Bouma, G. (1992). Feature structures and nonmonotonicity. *Computational Linguistics 18*(2), 183–204.

Carpenter, B. (1992). Skeptical and credulous default unification with applications to templates and inheritance. In T. Briscoe, A. Copestake, and V. de Paiva (Eds.), *Default Inheritance within Unification-Based Approaches to the Lexicon*. Cambridge: Cambridge University Press.

Frank, A. (1994). Verb second by lexical rule or by underspecification. Technical report, Institute for Computational Linguistics, Stuttgart.

Hinrichs, E. and T. Nakazawa (1994). Linearizing AUXs in German verbal complexes. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-driven Phrase Structure Grammar*, Lecture Note Series, pp. 11–38. Stanford: CSLI.

Hinrichs, E. and T. Nakazawa (1996). Applying lexical rules under subsumption. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, Copenhagen, pp. 543–549.

Iida, M., C. Manning, P. O'Neill, and I. Sag (1994). The lexical integrity of Japanese causatives. Paper presented at the LSA 1994 Annual Meeting.

Kathol, A. (1994). Passive without lexical rules. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-driven Phrase Structure Grammar*, Stanford, pp. 237–272. CSLI.

Lascarides, A., T. Briscoe, N. Asher, and A. Copestake (1996). Order independent and persistent typed default unification. *Linguistics and Philosophy 19*(1), 1–89.

Manning, C. and I. Sag (1995). Dissociations between argument structure and grammatical relations. Draft, Stanford University, July 1995.

Manning, C., I. Sag, and M. Iida (1996). The lexical integrity of Japanese causatives. In T. Gunji (Ed.), *Studies on the Universality of Constraint-based Phrase Structure Grammars*. Osaka University.

Meurers, W. D. (1995). Towards a semantics of lexical rules as used in HPSG. In *Proceedings of the Conference on Formal Grammar*, Barcelona.

Meurers, W. D. and G. Minnen (1995). The covariation approach as computational treatment of HPSG lexical rules. In *Proceedings of the Fifth International Workshop on Natural Language Understanding and Logic Programming*, Lisbon.

Miller, P. (1992). *Clitics and Constituents in Phrase Structure Grammar*. New York: Garland.

Pollard, C. and I. Sag (1987). *Information Based Syntax and Semantics, Volume 1*. Center for the Study of Language and Information Stanford.

Pollard, C. and I. Sag (1994). *Head-driven Phrase Structure Grammar*. Center for the Study of Language and Information Stanford.

Sag, I. (1995). Constraint-based Extraction (Without a Trace). Draft, Stanford University, November, 1995.

Sag, I. and P. Miller (to appear). French clitic movement without clitics or movement. *Natural Language and Linguistic Theory*.

van Noord, G. and G. Bouma (1994). Adjuncts and the processing of lexical rules. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*, Kyoto, pp. 250–256.

Filtering Left Dislocation Chains in Parsing Categorial Grammar

Crit Cremers*
Maarten Hijzelendoorn*

Abstract

This paper reports on a way to reduce the complexity of the process of left dislocation (re)construction for categorial grammar in the case of lexically assigned gaps, as an additional restriction on the complexity arising from lexical polymorphism in general. Specifying extraction sites lexically has the advantage that the combinatory explosion can be contained in the preparsing track by a specialized constraint on the expansion of sequences of categories. This constraint is called the *Left Dislocation Chain Filter* and is implemented by a Finite State Transducer. It is shown that the Filter can reduce the number of full string assignments under consideration prior to parsing with an average of one half to one order of magnitude, depending on the nature of the sentence.

1 Parsing Left Dislocation

Left dislocation is a very common, almost universal phenomenon in natural languages. It establishes the relation between an element at the left periphery of a clause and a particular, lexically open position in its right context. The leftward nature of dislocation is explained in Kayne (1994). The most prominent of these relations are invoked by so called wh-elements at the leftmost edge of questions and relative clauses. If a language has left dislocation, however, many other constituents can occur in a left dislocated position. Here are some examples from Dutch; the distinguished position in the right context is marked by *t*.

(1) Mimi vroeg zich af [wie] Jan dacht dat *t* zou gaan winnen
Mimi wondered (herself) who Jan thought that would go win
'Mimi wondered who Jan thought would win'

(2) [De film [die] ik heb *t* gezien] kan jij niet *t* gezien hebben
The movie that I have seen can you not seen have
'You cannot have seen the movie I have seen'

---

*Department of General Linguistics, Leiden University

In (2) we see two dislocations, one (a wh-induced type) within the boundaries of the other (a fronting of a 'normal' constituent). We will call the structure relating the left dislocated constituent and the empty position a left dislocation chain. We will refer to the two positions involved as the landing and the launch site, respectively, pursuing the dislocation metaphor. The lexical material at the landing site is also often referred to as the 'filler', and the other position as the 'gap'.

The properties and parameters of left dislocation chains are surely among the major topics of syntactic research in our days. It has become abundantly clear that there are major restrictions on these chains - represented by weak and strong islands for extraction - although there are many positions that may be part of one.

Natural language grammar is supposed to establish left dislocation chains, as the interpretation of left dislocated constituents is determined by their chain. Parsing natural language grammars therefore involves the (re)construction of left dislocation. The nature of this construction will correlate to the grammar's specification of dislocation, but the problem of parsing dislocation is quite general. At least the launch site of a chain is not explicitly marked - leaving aside prosodic information - and often, the lexical material in the landing site is not recognizable as being dislocated during lexical look-up. Consequently, a parser must actively compute the left dislocation chain in accordance with the grammatical nature of the relation; Van de Koot (1990) e.g. has an insightful treatment of the computational problem of left dislocation for Marcus-parsing. The need for the computation is evident: if a parser deduces that a certain noun phrase may be left dislocated, it has to check the possible noun phrase positions in the right context for being the launch site. This kind of parsing problem does only occur if the left-peripheral constituent of a clause is selected by an entity to its right. Adverbial adjuncts, for example, do not necessarily belong to this class of chain inducing entities: they are modifying other constituents, rather than being selected by them. One might, however, find good reasons, along with Bouma and Van Noord (1994), to consider adjuncts as arguments after all. In that case, their occurrence at the left periphery of a clause must be seen as dislocation and gives rise to chains that have to be computed as well. The present study is not biased with respect to this alternative.

In categorial grammar, one can think of at least two ways of establishing left dislocation chains. For categorial grammars exhibiting a full hypothetical logic, the 'gap' is constructed by withdrawing a hypothetical occurrence of a category and thus, by introducing a complex category; this approach is pursued e.g. in Hepple (1990) and Morrill (1994; ch.8).

Alternatively, gaps can be introduced as elements of lexical categories which are related to the filler by some form of 'gap threading' (cf. Pereira and Shieber 1987). This is the approach taken in DELILAH, a grammar/parser system for Dutch developed by the authors. In the latter system, gaps are introduced as such in lexical assignments of categories to words, alternating with assignments providing lexical arguments (see below). In any case, the parser of a categorial grammar has to check several filler-gap combinations in a trial-and-error mode in order to determine a left dislocation chain. This paper reports on a way to reduce the complexity of the process of left dislocation (re)construction for categorial grammar in the case of lexically assigned gaps, as an additional restriction on the complexity arising from lexical polymorphism in general.

2 Lexical Ambiguity and Parsing Categorial Grammar

Lexical ambiguity is known to be a major threat to efficient parsing of natural language. Barton, Berwick and Ristad (1987: ch.3) demonstrate that the combination of simple agreement and lexical ambiguity makes natural language parsing NP-complete, i.e. a standard problem in the class of computationally intractable problems. Evidently, agreement can be taken to go proxy for all kinds of mutual dependencies between phrases in a sentence. Dependencies like agreement are at the heart of natural language, and adequate grammars must account for it. As a consequence, an adequate grammar of natural language can hardly be parsed efficiently if it has to allow for lexical ambiguity. In this vein, Johnson (1991) proves that even the Tomita algorithm for generalized LR parsing shows exponential complexity when applied to lexically ambiguous grammars. In this proof, the exponent is determined by the size of the grammar. These results confirm the observation in Gazdar and Mellish (1989: p.169) that "The cubic worst-case time efficiency problem for natural language parsers (...) is completely dwarfed in practice by a much more serious problem, that of pervasive natural language ambiguity".

Unfortunately, this statement is fully applicable to categorial grammar. Categories can be seen as combinatorial agendas. Every category imposes a set of requirements on its context. A string of categories represents a well-formed sentence only if these requirements turn out to converge. A certain degree of lexical ambiguity – or rather: polymorphism *per* lexical atom – seems inevitable. In categorial grammar, for example, differences in subcategorization (a verb may select an infinitival as well as a tensed complement), word order (a finite verb may have its complements to the left or to the right) or double functionality (a word might be a preposition or a particle) must lead, in some stage of the parsing process, to branching possibilities and an increase of search space. As an example of a lexical item which introduces many combinatorial agendas, consider the case of Dutch *willen*, 'to want'. The lexicon of Dutch has to specify for *willen* at least the following different categories, which are casted here in a neutral format:

(3)

(i)	vp/vp	(infinitival form with vp-complement)
(ii)	vp/s_sub	(infinitival form with tensed sentential complement)
(iii)	s/vp/np	(finite form with vp-complement for verb-second and verb-first main sentences)
(iv)	s/s_sub/np	(as (iii), but with sentential complement)
(v)	s_vn\np/vp	(as (iii), but for verb-final sentences)
(vi)	s_vn\np/s_sub	(as (v), but with tensed complement)

This list is not necessarily complete. For example, if one needs to distinguish declarativity from other sentential modes like questioning, more sentential categories may be added. The variety that arises, cannot be handled by type changing rules. As a matter of fact, none of the types listed in (3) can be deduced from another type in the list by canonical type changing rules, though every finite type is a regular and predictable expansion of one of the basic infinitival types.

In general, we can describe the problem of lexical ambiguity for categorial grammar as follows. Let G_{NL} be a categorial grammar for a language NL, and L a lexicon with initial assignment $A(w_i)$ of nonterminals to the words w_i of NL. For example, A(*willen*) contains at least the categories given in (3). Let $S = w_1 \dots w_n$ be a sentence over L. Deciding whether S is in NL amounts to searching some sequence C $= c_1 \dots c_n$, with $c_i \in A(w_i)$ such that C is derivable under G_{NL}. The solution to the problem may require checking the derivability of many such Cs. Basically, for a certain S the number of sequences the derivability of which must be checked is $\Pi_1^n |A(w_i)|$, the Cartesian product over w_i, which is exponentially dependent on n. $\Pi_1^n |A(w_i)|$ defines the search space for parsing S. The search space should not be defined by spurious ambiguity, however. It makes sense to require for each c in the lexical assignment of some word w that there is a sentence in NL containing w, which can only be derived under G_{NL} if c is in $A(w)$. In this vein, we require every initial assignment of a category to a word to be necessary with respect to G_{NL}. Note that the search space is not influenced by the algorithm of the theorem prover itself. One can, however, represent lexical ambiguity as a complex category by means of additional type constructors, as is suggested e.g. by Morrill (1994: ch.6). To an ambiguous term a conjunction of categories and/or a disjunction of arguments is assigned. At each deduction step in which this complex category is involved, the theorem prover has to choose which of the coordinated types is activated. No hope is offered, though, as to the efficiency of this procedure.

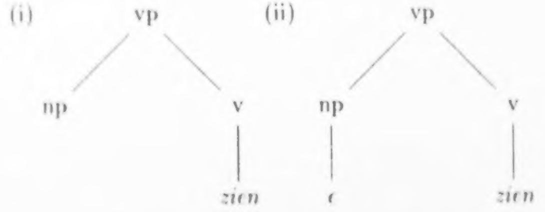
This approach to ambiguity shows, however, that there is, to a certain degree, a trade-off between lexical ambiguity and properties of the grammar. In particular, a grammar may assign nonterminals to certain lexically assigned nonterminals, by having monadic rules or theorems of the type $c \rightarrow c'$. One can think of the famous lifting rule $x \rightarrow y\backslash(y/x)$ and the Geach rule $x/y \rightarrow (x/z)/(y/z)$. Their presence, however, causes a serious problem for theorem proving itself, as they induce spurious ambiguity (see e.g. Wittenburg 1986, König 1990, and Hepple 1990 for analyses and performance-oriented remedies of this phenomenon). In these cases, the search space for parsing S is partially constructed by the grammar itself. Again, the question whether G_{NL} derives S, induces an explosion of queries as to whether G_{NL} derives a certain C.

Because of the logic of categorial grammar, theorem proving is a genuine model for parsing these grammars (Hepple 1990). Propositions, which have to be checked for being a theorem, are sequences of categories. If a sentence gives rise to more than one sequence, all these sequences - i.e. all these combinations of combinatorial agendas - have to be checked. Of course, the theorem prover itself can be trimmed in several fashions, pertaining on the complexity of the proof-finding process. If, however, the categorial grammar involved is of a (mildly) context-sensitive nature (as are the Combinatory Categorial Grammar of Steedman (1990) - see Joshi *et. al.* (1991) - and the grammar in the present DELILAH system - see below) the theorem prover is confronted with the fact that context-sensitive recognition is PSPACE-complete (Hopcroft and Ullman 1979; ch.13). But the major burden on the parsing process is imposed by the multiplicity of potential theorems, independently of the properties of the deductive system. Below we will discuss how the combinatorial explosion of hypotheses can be controlled in DELILAH.

Here we will concentrate on the processing of a particular source of lexical polymorphism: the possibility that in a locally-defined argument structure one argument may be missing as a result of left dislocation, also known as movement to [Spec, CP]. The categorial lexicon of Dutch may specify as a category of the verb *zien* 'to see' not only $vp\backslash np$ to indicate that it is meant to head a configuration with an *np* to its left, but also $vp\backslash np^\wedge gap$ to indicate that that object may not be present.

If we consider the category $vp\backslash np$ as introducing a local tree of type (4)(i), to which a local tree with a root *np* can be adjoined at the node marked as such, the category $vp\backslash np^\wedge gap$ must be seen as the introduction of the local tree (4)(ii) in which this node is barred by emptiness. Structurally, the trees are the same, though they will be treated differently by the rules of grammar (see below).

(4)



Gapped categories are invariantly specified as left arguments, since dislocation is leftward. Therefore, the gapped counterparts to the categories pp/np and vp/vp are $pp\backslash np^\wedge gap$ and $vp\backslash vp^\wedge gap$, respectively.

The additional specification of *zien* as a verb that may lack an adjacent object is predictable – the objects of transitive verbs are generally available for extraction – but not trivial. Not all *np* arguments occurring in some lexically assigned category are candidates for extraction; the *np* in a possessive determiner (*np*'s *n*), for example, is not. Moreover, it is useful to store the information that, within a certain complex category, at most one argument is available for extraction. The verb *geven* 'to give' has one category specifying a bitransitive infinitive, but also two additional categories specifying the separate extractability of each argument of that infinitive – only one argument can be dislocated, of course. Consequently, in infinitival position *geven* has three lexical options, instead of one; the number of finite lexical categories associated with *geven* multiplies accordingly.

The specification of extractability, then, may increase the number of categories assigned to a particular lexical item and contribute to a search space explosion for parsing. As noted above, one could choose not to specify launching sites lexically, but to compute the possibilities while parsing. For Lambek's categorial grammar, there is the option of hypothetical reasoning, as in Morrill (1994), and in other frameworks one can implement other forms of gap threading, as in Stabler (1992). But deriving all possible extraction sites is not necessarily more efficient than specifying them. In fact, we will show that specifying extraction sites lexically has the advantage that the combinatory explosion can be contained in the pre-parsing track by a specialized constraint on the expansion of sequences of categories. Moreover, the lexical approach complies with the argument by Johnson and Kay (1994) that gap hypotheses in a derivation must be licensed by lexical items in order to assure termination of the parsing process.

To a large extent, the art of parsing is finding secure means to restrict the number of possible assignments. Given the exponential function $\Pi_1^n |A(w_i)|$, efficiency requires serious pruning of the search space. Optimally, the means to achieve this are anchored in the grammar that is to be applied. Resource-sensitive categorial grammar offers some options for pre-checking assignments. The parsing system DELILAH incorporates an instance of a bracket-free, mildly context-sensitive categorial grammar. It deals with various forms of discontinuity, like free coordination, verb raising and long distance dependencies.

The grammar basically consists of one cancelling operation, generalized composition (cf. Steedman 1990, Joshi. *et al.* 1991), operating on two triples of the form Head \ LeftArgumentList / RightArgumentList.

Heads are basic types; they may be cancelled while the argument lists of their categories merge with the argument lists of the category that provokes the cancelling. The formalism and its applications are discussed in Cremers (1993; ch.2). A related, but slightly less expressive formalism is defined in Milward (1995) as AB Categorial Grammar with Associativity (AACG). DELILAH's grammar may be taken to have at least mildly context-sensitive power, as it extends the concept of generalized composition, which Joshi *et al.* (1991) prove to be in that class.

In (5) we present the general scheme for a left-to-right cancelling. (Of course, right-to-left cancelling also exists.)

(5) *DELILAH's Grammar Format*

If a string with category

PrimaryHead \ LeftList / [SecondaryHead^Operator | RestRightList]

occurs to the left of a string with category

SecondaryHead \ OtherLeftList / OtherRightList,

combine these strings – under some restrictions triggered by Operator with respect to the content of the argument lists – to a string of category

PrimaryHead \ NewLeftList / NewRightList,

where NewLeftList and NewRightList stem from appending the two left lists and the two right lists, respectively, in either one of two possible orders, which encode either continuity or discontinuity.

In a rule notation we get (6), where *append*$_o$(L1,L2) is some append-operation triggered by some operator ^*o*, yielding a list whenever it is defined for L1 and L2, and non-executable otherwise. In the latter case, the two categories to the left of the arrow cannot reduce to one by cancellation of Sec.

(6) Prim\LList1/[Sec^o|RList1] Sec\LList2/RList2 →
  Prim\append$_o$(LList1,LList2)/append$_o$(RList1,RList2)

A string is considered to be a well-formed sentence, iff the lexical hypotheses (categorial agenda's) can be reduced by recursive applications of (6) to one category *s\[]/[]*. This grammar strictly preserves directionality (cf. Steedman 1990), only cancels elementary types, and does not use hypothetical reasoning. Instead, because the composition rule takes into account the full internal structure of both the primary and the secondary category, non-peripheral extraction can be treated without specialized operators like the up and down arrows of Moortgat (1988). Dislocation and other forms of word order variation can be handled by composition alone. (7) shows two options of adjunction to a verb; both options are available if the operator ^*o* is defined for the relevant internal structure of the secondary category at the stage of cancelling *vp*.

(7) (i) vp\[]/[vp^o] np\[]/[] vp\[np^-]/[] ⇒
  vp\[]/[vp^o] vp\[]/[] ⇒
  vp\[]/[]
 (ii) np\[]/[] vp\[]/[vp^o] vp\[np^-]/[] ⇒
  np\[]/[] vp\[np^-]/[] ⇒
  vp\[]/[]

The combination of directionality and the fact that the grammar only cancels basic types - as does Milward's (1995) AACG - has interesting consequences for parsing. For a given prefix $P(i)$ of assignments to the first i words of a sentence, we can decide whether it makes sense to add to it a certain lexical category of the $i+1$th word, that is, we can decide whether $P(i)$ plus that certain category can be the prefix $P(i+1)$ of a sequence of categories which may be parsed succesfully. If not, that particular extension of $P(i)$ is rejected, and with it all the (virtual) sequences in the set with cardinality $\Pi_1^n |A(w_i)|$ that have it as a prefix. Technically, these prefixes are best considered to be paths of a tree which is built tier-by-tier during lexical look-up: a directed acyclic graph with categories as vertices and edges between neighbours in a sequence. At the extreme vertex of every path, information is accumulated on the type pattern of the path. A new category is added as a vertex connected to a path and extending that path only if its agenda is not incompatible with the information at the 'preceding' vertex. When the category is added as a new vertex, it stores the updated information on the extended path. If no category of a certain word is compatible with an existing path, the path is pruned; all remaining (active) paths are of equal length. The main instrument here is an operationalization of Count Invariance (Van Benthem 1986, Moortgat 1988, König 1990): the property that sequences of complex types can be derived only if they exhibit a certain balance of primitive types. This is the central strategy used in DELILAH to restrict the search space, even in the context of coordination (see Cremers and Hijzelendoorn 1997 and Cremers 1989). Thus, the search space is considerably limited *on-line*. As a matter of fact, while building the tree, the pruning rate exceeds the growth factor of the search space (cf. Cremers and Hijzelendoorn 1997). As $\Pi_1^n |A(w_i)|$ explodes with n, the proportion *#active-paths-of-length-n* $/\Pi_1^n |A(w_i)|$ decreases exponentially. The decrease of this proportion justifies the construction of the pruned tree of viable prefixes-up-to-n. For a numerical illustration of this effect, consider the parsing of the sentence

(8) Wie zegt de man, die ik wilde laten werken, dat hem gedwongen heeft mij met de poppen te laten spelen?
Who says the man that I wanted let work that him forced has me with the dolls to let play?
'Who does the man, I wanted to work, say that forced him to let me play with the dolls?'

Under DELILAH's present lexicon, the search space $\Pi_1^n |A(w_i)|$ for this sentence consisting of 20 words contains 822,528,000 possible assignments (paths). Building and pruning the tree with a remainder of 109 paths takes 6,120 ms cpu time (excluding time for garbage collecting, stack shifting, or in system calls) on a Silicon Graphics Indigo R4000 workstation; this includes the time taken by the special pruning algorithm for gapped categories to be discussed below. Parsing these 109 non-rejected sequences takes 360 ms, say 3 ms for each. Since there is no principal difference between paths which were pruned and paths which survived pruning, we can deduce that parsing the whole tree of sequences would have taken 822,528,000 times 3 ms is 2,467,584,000 ms. This is about 411,000 times as much as DELILAH needed to construct-and-prune the search space.

It is worth noting that the pruning does not put any claim on the parsing strategy that is applied. The dynamic application of Count Invariance only selects what has to be parsed, not how this task is performed. In particular, since not all the remaining paths will differ from each other at any node, one can think of a form of chart parsing to exploit the remaining hypotheses space. On the other hand, little is known about efficient parsing of context-sensitive grammars.

Under the grammar sketched above, left dislocation is solved syntactically by bringing together the left dislocated constituent and a unique gap. The gap is transported leftwards by generalized composition: in fact, a gap is 'inherited' by every category of every string that contains the word introducing the gap. Finally, the gap is the only left argument of the constituent to the right of the dislocated phrase. In the example derivation (9) with a left dislocated noun phrase, X and Z are sequences of categories. Only a few stages of the derivation are made explicit; they are connected by subsequent application of generalized composition (5–6).

```
(9) np\[]/[]  s\[]/[...]  X  y\[...]/[vp^o]  vp\[...,np^gap]/[...]  Z  ⇒
    np\[]/[]  s\[]/[...]  X  y\[...,np^gap]/[]  ⇒
    np\[]/[]  s\[]/[s^o]  s\[np^gap]/[]  ⇒
    np\[]/[]  s\[np^gap]/[]  ⇒
    s\[]/[]
```

Except for special circumstances, which we will not consider here (e.g. parasitic gaps; for a treatment see Morrill 1994), gaps and dislocated constituents are related one-to-one.

3 Filtering Left Dislocation Chains

Although Count Invariance (exploiting the resource sensitivity of certain categorial systems) is operationalized in DELILAH for on-line reduction of search space prior to proper parsing, it cannot discriminate between configurations (possible prefixes of sequences of lexical assignments) other than by the mere occurrence of basic types. Gaps have to be marked in the lexicon for the category they represent. If a gap is to be bound to a preposed *np*, it has to be marked for this binding as a gap or variable of that particular type. Consequently, the introduction of gaps may increase the number of categories of a certain lexical item that looks for a particular type to the left; gaps are always bound by left dislocated constituents (cf. Kayne 1994). As for the application of Count Invariance, there is no difference between a category *vp\[np]/[]* and its gapped relative *vp\[np^gap]/[]*: they expand the same tree in terms of number, labels, and structure of nodes (cf. (4)). If Count Invariance allows for the attachment of one of them to a prefix $P(i)$, it also allows for the attachment of the other. In this case, the number of sequences to be checked for grammaticality is doubled by the mere presence of a gapped category in the lexicon for the $i + 1$th word in the sentence. In general, Count Invariance is underspecified with respect to the grammatical extension of prefixes, as it is applied prior to parsing.

The main contribution of gapped categories to the extension of the search space is caused by the gap's location being undetermined. In a given string of words, there are many possible candidates that introduce a gapped category, though in the final parse only one of the candidates will turn out to be the real gapped category. Sequences like the following are hardly candidates for succesful parsing if only one of the categories introduces a finite domain.

```
(10) np  ...  s\[np^gap]/...  ...  vp\[np^gap]/...  ...
```

It makes sense, then, to look for additional means to prevent spurious accumulation of gapped categories in a sequence, just to accommodate the gap's indeterminacy *a priori*.

For every prefix of a sequence of lexical categories, we use a Finite State Transducer (FST) in DELILAH to keep track of the maximal number of gaps that must be made available in the suffix of that sequence. This FST performs its tasks in the very same pre-parsing track in which Count Invariance is used dynamically to prune the search space. The timing for the pre-parsing procedure for (8) given above, included the operations of the FST.

The general idea is as follows. Every prefix $P(i)$ of a sequence of assignments S(n) is associated deterministically with a state of an FST and a number generated by that state. Recall that these prefixes can be seen as paths of a tree under construction. The number that the FST provides, is associated with the extreme vertex of that path. This number indicates how many gap 'slots' are available for a suffix to that prefix. The states of the FST represent the relevant features of a particular $P(i)$. The $i + 1$th word may introduce a category for which the FST is defined in that state, or not. If the category is in the domain of the state of the FST with which $P(i)$ is associated, it will force the FST to move. This move may involve adding 1 or subtracting 1 from the counter, or even a reset of the counter. Addition moves are forced by types which introduce a syntactic domain from which extraction is permitted. Finite verbs, for example, force addition moves. Subtraction moves are forced by categories containing gaps, or – in certain states – by categories that indicate closure of extraction domains. No subtraction move can be made if the counter is zero. In that case, the FST fails, and the category will not be added to $P(i)$ to form $P(i + 1)$, since $P(i + 1)$ cannot be associated with a state of the FST. The category may be added to some other prefix $P(i)$, though. Also, another category from the $i + 1$th word's lexical assignment may be added to $P(i)$ to form a $P(i + 1)$.

Thus, the moves of the FST are triggered by types occurring in a category of the $i + 1$th input word for a given prefix of assignments $P(i)$. In its actual form, DELILAH activates the FST only if the category that is a hypothetical extension of a prefix of a sequence of assignments is either 1. headed by a main sentential type, 2. has a finite sentential type as an argument, 3. contains a gapped type, or 4. is headed by prepositions. The first two triggers will be evident: they are the ones that indicate finite domains, the main extraction fields. They force the FST to make an addition move. The third one is also evident: it is the one that forces a subtraction move. PP's need a special treatment in order to account for certain consequences of pied-piping. They have to make available an additional gap option, apart from the option that allows for their own dislocation which is introduced in a standard way.

Here is a description of the FST and some comments. The triggering types are:

head_main_s (hms)	the type that is the head of a finite verb in main sentences; e.g. s in $s\backslash[np^{\wedge}gap]/[vp]$
head_embedded_s (hes)	the type that is the head of a finite verb in embedded sentences; e.g. s_vn in $s\_vn\backslash[np\ np]/[\ ]$
head_pp (pp)	the head of a prepositional category; e.g. pp in $pp\backslash[\ ]/[np]$

right_argument_embedded_s (raes)	the type that is a right hand side argument of a complementizer, announcing the start of an embedded sentence; e.g. s_vn in $np\backslash[np]/[s\_vn]$
gap (gap)	the left hand argument of any category that is marked for gap; e.g. $pp^\wedge gap$ in $vp\backslash[np\ pp^\wedge gap]/[\ ]$

Of each type, the head is processed first, then its left searching arguments, and finally its right searching arguments. The transition scheme is given in (11) as a relation between a triple <state, type, number> and a pair <state, number>. Some very idiosyncratical transitions are left out for transparency reasons.

(11) *Left Dislocation Chain Filter*

initialization: <a, 0>	
<a, hms, X> ⇒ <e, 1>	finite main verb resets gap options
<a, hes, X> ⇒ <a, X>	finite embedded verb does not change options; necessary for coordinated structures
<a, gap, X> ⇒ <a, X−1 > if X > 0	consumption of gap option
<a, raes, X> ⇒ <b, X+1 >	introducing a new domain for dislocation
<b, raes, X> ⇒ <b, X+1 >	idem
<b, hes, X> ⇒ <b, X>	as before
<b, gap, X> ⇒ <b, X−1 > if X > 0	as before
<c, hms, X> ⇒ <e, 1>	all options not yet consumed are lost; number of options reset to one
<c, hes, X> ⇒ <d, X>	just a state transition
<c, raes, X> ⇒ <b, X+1 >	as before
<c, gap, X> ⇒ <c, X−1 > if X > 0	consumption of gap option
<d, hes, 0> ⇒ <a, 0>	re-initialization; the end of a domain is introduced; all options used
<d, hes, X> ⇒ <c, X−1 >	end of domain; gap option for that domain not used; number of options decreases
<d, raes, X> ⇒ <b, X>	idem, but also start of new domain: options not changed
<d, hms, X> ⇒ <e, 1>	as before
<d, gap, X> ⇒ <c, X−1 > if X > 0	consumption
<e, raes, X> ⇒ <b, X+1 >	introduces new domain with additional option
<e, gap, 1> ⇒ <a, 0>	consumption
<S, pp, X> ⇒ <[S], X+1 >	for every state S a special state [S] is needed for pied-piping phenomena, introducing an additional gap option
<[S], gap, X> ⇒ <S, X−1 >	consumption in 'pied-piping' state
<[S], ε, X> ⇒ <S, X−1 >	empty move otherwise; the extra option is cancelled

A category C will be added to a prefix of assignments $P(i)$ in state S with number N if the FST is defined in S for the types in C. A simple example is given in (12).

(12) Ik wil elke man een boek geven
I want every man a book give
'I want to give every man a book'

Let the string of categories in (13) be a prefix of assignments (one of possibly many more) to the first six words, up to *boek*, i.e. *P*(6).

(13) np s\[np^gap]/[vp] np\[]/[n] n np\[]/[n] n

That prefix, just one among others that may have survived thus far, will be associated with state <a,0>. This can be seen as follows. The FST is entered in state <a,0>. The first type, *np*, is not a triggering type; the FST does not move. The head of the second type, *s* is an hms; the FST moves to state <e,1>, creating an option for a gap to be consumed later. The left searching category *np^gap* is a gap, which brings the FST back to state <a,0>. The right searcher *vp* has no effect, because it is not a triggering type. The same is true for the heads and searchers of the third, fourth, fifth and sixth type in (13). Now *geven* will be considered. Among its lexical categories we find some that contain gaps, like *s*\[*np^gap np np*]/[ ] and *vp*\[*np np^gap*]/[]. Most of them will be rejected for concatenation to that prefix, since a category containing a gap cannot be processed from <a,0>. The only option for a gapped category of *geven* would be one that is headed by a finite main sentence type (hms), for this category would bring the FST in state <e,1>, introducing a gap option. This option is rejected, however, because of other filtering mechanisms apart from the left dislocation FST. Only categories headed by *vp* with no gapped argument remain as possible continuations of the prefix. The number of assignments that has to be parsed (checked for derivability) is seriously cut down.

4 The Effect of Chain Filtering

We have tested the effect of chain filtering by computing three values for a certain set of sentences:
(a) the number of full string assignments left under chain filtering
(b) the number of full string assignments left without chain filtering
(c) the number of full string assignments in case the lexicon would have no gaps.
In the latter case chains must be identified by hypothetical reasoning. In our approach gaps are fixed per full string assignment: no additional hypotheses as to the possible occurrences of gaps are necessary while parsing.

In (figure 1) at the end of the paper the table of results is given; the results are presented as natural logarithms to express their order of magnitude. They are ordered with respect to the length of the sentences in the test set (column a). All counts are submitted to a cluster of other filtering devices which are not related to left dislocation, but which do a major job at distinguishing viable from inviable prefixes, as was discussed for example (8). The numbers of full string assignments, which survived the general filtering devices including and excluding the Chain Filter, mark a rather undetermined stage in the processing of the sentences; the strings counted below are not necessarily all parsed. Most of the sentences are coordinated sentences, which complicates any filtering of prefixes: the selection is *pre*-parsing, the coordinates are not yet determined at that stage, and this indeterminacy must be reflected in weakened application of the FST (which is not spelled out above).

The exponents given in column (f) of the table hold the main result of chain filtering. They indicate the difference between the number of sequences of categories that must be processed if chain filtering is applied (column d) and the number of sequences of categories that must be processed if chain filtering is not applied (column e). Both numbers can be compared to the number of sequences that would survive Count Invariance if the lexicon would not contain gapped categories (column c). In that case, however, DELILAH will not be able to parse left dislocation any longer. Column (b) lists the Cartesian product over w_i, i.e. without any filtering, but including gapped categories. It defines the upper bound for the exponents in the columns (c), (d) and (e).

From these figures one can see that the Left Dislocation Chain Filter has a measurable effect on the number of sequences that must be parsed. It can reduce the number of full string assignments under consideration prior to parsing with an average of one half to one order of magnitude (column f), depending on the nature of the assignments, i.e. the nature of the sentence. In the final case, for example, the application of the Chain Filter reduces the number of full string assignments at this particular stage of processing by a factor of $e^{1.39} = 4$. All possible analyses are kept in store, however; in that respect, chain filtering is as conservative as necessary.

The number of sequences left after chain filtering is still considerably larger than the number a parser checking dislocation by hypothesising gaps would have to consider. This is not surprising. Specifying gaps lexically introduces at least n additional sequences for every gap-less sequence of assignments with n extractable arguments. It depends on the parsing procedure to what extent this complicates the parsing of the sentence. DELILAH can parse these additional assignments marked for gaps deterministically: the number of assignments is the only factor affecting the complexity of the solution to the problem of left dislocation.

It is by no means clear that the Left Dislocation Chain Filter is stated in the best possible way. It must be stressed, however, that in the presence of coordination – all but three of the sentences measured above are coordinated ones – filtering left-dislocated chains is weakened by necessary precautions with respect to across-the-board phenomena. As long as one does not know what is coordinated exactly, the substring to the right of a coordinating element may have to accommodate all the chains that were possibly established at the left of the coordinator. In this respect, the results show that chain filtering keeps performing under difficult conditions.

Acknowledgements

We would like to thank two anonymous reviewers for their valuable comments, Rob Goedemans for checking our English, and Ton van der Wouden for translating the text into LaTeX.

The system DELILAH is available at
http://fonetiek-6.LeidenUniv.nl/hijzlndr/delilah.html.

column a: sentence length in words
column b: natural logarithm (*ln*) of the number of full string assignments (unfiltered; $\prod_1^n |A(w_i)|$) for a lexicon with gapped categories
column c: (*ln* of the) number of full string assignments (filtered by independent checks) for a lexicon without gapped categories
column d: (*ln* of the) number of full string assignments (filtered) for a lexicon with gapped categories, and **with** application of chain filtering
column e: (*ln* of the) number of full string assignments (filtered) for a lexicon with gapped categories, and **without** application of chain filtering
column f: difference d − e; effect of chain filtering, in orders of magnitude; a negative effect means reduction of the search space

a	b	c	d	e	f
#words	Cart. product over w_i +gapped cats.	#assignments -gapped cats. -Chain Filter	#assignments +gapped cats. +Chain Filter	#assignments +gapped cats -Chain Filter	Chain Filter effect
7	1.60	0.69	1.09	1.09	0.00
8	4.02	2.07	3.17	3.17	0.00
9	9.91	4.14	6.76	6.83	-0.07
10	4.02	2.07	3.17	3.17	0.00
11	11.38	0.0	2.07	2.77	-0.70
13	10.13	3.17	5.41	5.77	-0.25
14	14.02	3.17	7.32	7.57	-0.25
15	6.10	3.46	4.85	4.85	0.00
16	15.02	4.27	8.58	8.95	-0.37
17	14.45	3.87	7.76	8.41	-0.65
18	18.65	1.79	5.54	7.56	-2.02
19	17.44	5.25	9.77	10.22	-0.45
20	16.60	5.06	10.06	10.63	-0.57
21	13.68	4.85	9.63	10.02	-0.39
22	20.06	3.46	9.10	10.58	-1.48
23	15.83	4.56	9.36	9.80	-0.44
24	8.05	3.46	6.64	6.64	0.00
25	18.31	7.09	13.66	13.99	-0.33
26	9.57	4.85	8.72	8.72	0.00
28	21.03	5.95	12.07	12.58	-0.51
29	26.36	5.50	13.10	14.08	-0.98
32	29.43	1.09	7.24	8.65	-1.41
34	23.12	5.25	12.79	13.64	-0.85
37	28.18	4.85	13.23	14.62	-1.39

Figure 1: **Table of Results**

References

Barton, G., R. Berwick, and E. Ristad (1987). *Computational Complexity and Natural Language*. MIT Press.

Benthem, J. v. (1986). *Essays in Logical Semantics*. Reidel.

Benthem, J. v. (1991). *Language in Action*. North-Holland. SLFM 130.

Bouma, G. and G. van Noord (1994). Constraint-based categorial grammar. In *Proceedings 32nd Annual Meeting of the ACL*, pp. 147–154. ACL.

Cremers, C. (1989). Over een lineaire kategoriale ontleder. *TABU 19*(2), 76–86.

Cremers, C. (1993). *On Parsing Coordination Categorially*. Ph. D. thesis, Leiden University. HIL dissertations 5. Also available at ftp://fonetiek-4.LeidenUniv.nl/pub/cremers/dissi.ps.

Cremers, C. and M. Hijzelendoorn (1997). Pruning search space for parsing free coordination in categorial parsing. To appear in: *Proceedings International Workshop on Parsing Technologies*, MIT 1997.

Gazdar, G. and C. Mellish (1989). *Natural Language Processing in PROLOG*. Addison-Wesley Publ Cy.

Hepple, M. (1990). *The Grammar and Processing of Order and Dependency*. Ph. D. thesis, University of Edinburgh.

Hopcroft, J. and J. Ullman (1979). *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley Publ Cy.

Johnson, M. (1991). The computational complexity of glr parsing. In M. Tomita (Ed.), *Generalized LR Parsing*, pp. 53–42. Kluwer.

Johnson, M. and M. Kay (1994). Parsing and empty nodes. *Computational Linguistics 20*(2), 289–300.

Joshi, A., K. Vijai-Shanker, and D. Weir (1991). The convergence of mildly context-sensitive grammar formalisms. In P. Sells, S. Shieber, and T. Wasow (Eds.), *Foundational Issues in Natural Language Processing*, pp. 31–82. MIT Press.

Kayne, R. (1994). *The Antisymmetry of Syntax*. MIT Press.

König, E. (1990). *Der Lambek-Kalkul. Eine Logik für lexikalische Grammatiken*. Ph. D. thesis, Universität Stuttgart. IWBS Report 146.

Koot, J. v. d. (1990). *An Essay on Grammar-Parser Relations*. Ph. D. thesis, University of Utrecht.

Milward, D. (1995). Incremental interpretation of categorial grammar. In *Proceedings 7th EACL*, pp. 119–126. Dublin: EACL.

Moortgat, M. (1988). *Categorial Investigations*. Foris.

Morrill, G. (1994). *Type Logical Grammar*. Kluwer.

Pereira, F. and S. Shieber (1987). *Prolog and Natural Language Analysis*. CSLI.

Stabler, E. (1992). *The Logical Approach to Syntax*. MIT Press.

Steedman, M. (1990). Gapping as constituent coordination. *Linguistics and Philosophy 13*(2), 147–171.

Wittenburg, K. (1986). *Natural Language Parsing with Combinatory Categorial Grammar in a graph-unification-based Formalism.* Ph. D. thesis, University of Texas at Austin.

Speech Output Generation in GoalGetter

Esther Klabbers[*][†]

Abstract

In this paper a method for speech output generation in data-to-speech systems is proposed, called phrase concatenation, which tries to find a balance between naturalness and flexibility of the speech output. The GoalGetter system, which generates spoken monologues on football matches, serves as an example. The phrase concatenation technique involves concatenating prerecorded words and phrases, which is new in that different prosodic versions of otherwise identical phrases are recorded.

Introduction

The main issue addressed in this paper is the problem of generating high quality speech in data-to-speech systems, i.e., systems which present data in the form of spoken monologues, sometimes also called concept-to-speech systems. Data-to-speech generation is a relatively new area of research. Traditionally, research on spoken-language generation was mainly undertaken within the separate fields of natural-language generation and text-to-speech synthesis. State-of-the-art language generation is capable of generating flexible utterances and texts, but often the intonational properties are not taken into account. Text-to-speech synthesis often fails to generate adequate prosody due to the lack of information available in texts. In contrast to text-to-speech systems, explicit discourse models can be reliably constructed in data-to-speech systems, so that a more natural prosody can be achieved.

The method of speech output generation is explained in the context of a simple data-to-speech system called GoalGetter, which generates spoken monologues on football matches. GoalGetter generally works as follows: it takes as input a Teletext page that contains summary information on a particular football match. The Teletext page lists the two teams that played against each other, the score, which players scored when, etc. From this concise information, the language generation module (LGM) generates a coherent text using syntactic templates. The output text, enriched with prosodic markers, is passed on to the speech generation module (SGM), which makes it audible through one of two output modes, i.e., diphone synthesis or phrase concatenation.

---

[*] IPO, Center for Research on User-System Interaction

[†] This research is carried out within the framework of the Priority Programme Language and Speech Technology (TST). The TST-Programme is sponsored by NWO (Netherlands Organization for Scientific Research).

Before explaining the phrase concatenation technique, it is necessary to get a general idea of the working of the LGM. It is responsible for the content and form of the utterances and the prosodic properties, and as such sets the pre-conditions the SGM has to satisfy.

1 Language Generation in GoalGetter

The technique used for natural language generation in GoalGetter was originally developed at IPO for an English-spoken database query system called Dial-Your-Disc (DYD). This system generates spoken monologues about compact discs with musical compositions written by Mozart (van Deemter, Landsbergen, Leermakers, and Odijk 1994). The architecture of the LGM is depicted in Figure 1.

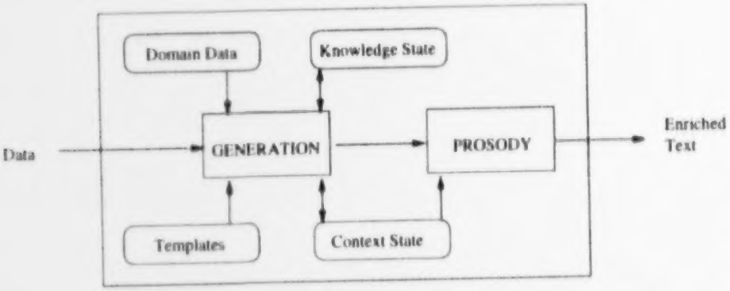


Figure 1: The architecture of the Language Generation Module (LGM)

team 1:	PSV
goals 1:	1
team 2:	Ajax
goals 2:	3
goal 2:	Kluivert (5)
goal 2:	Kluivert (18)
goal 2:	Blind (83/pen)
goal 1:	Nilis (90)
referee:	Van Dijk
spectators:	25 000
yellow 1:	Valckx

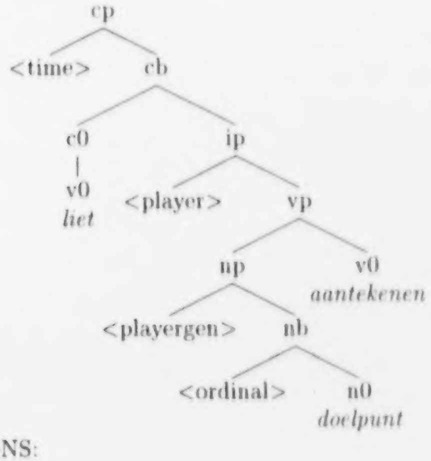
"De "wedstrijd tussen "PSV en "Ajax / eindigde in "@een // - "@drie /// "Vijfentwintig duizend "toeschouwers / bezochten het "Philipsstadion ///

"Ajax nam na "vijf "minuten de "leiding / door een "treffer van "Kluivert /// "Dertien minuten "later / liet de aanvaller zijn "tweede doelpunt aantekenen /// De % "verdediger "Blind / verzilverde in de "drieentachtigste minuut een "strafschop voor Ajax /// Vlak voor het "eindsignaal / bepaalde "Nilis van "PSV de "eindstand / op "@een // - "@drie ///

% "Scheidsrechter van "Dijk / "leidde het duel /// "Valckx van "PSV kreeg een "gele "kaart ///

Figure 2: Example input and output of the LGM

The input for the *Generation* module in the LGM is formed by a textual representation of a teletext page on a particular football match (see Figure 2). It also uses a database that contains fixed background data about e.g., the names of the stadiums and the field positions for each player (defender, goalkeeper). To generate sentences, the Generation module uses a set of so-called syntactic templates. These are basically syntactic parse trees with fixed parts, *carriers*, and variable parts, *slots*, in which other syntactic templates can be inserted. An example template is depicted in Figure 3. The templates have conditions attached to them about when they can be used. For instance, a template expressing the number of spectators of a match can only be used after the match was introduced, e.g. by naming both teams. In order to be able to check which information is already known, a *Knowledge State* is maintained. Furthermore, to ensure the well-formedness of referring expressions used to fill the template slots, we need information about which discourse objects have been mentioned, and how and when they have been referred to. This is recorded in the *Context State*. Each piece of information in the data structure can be expressed by at least one template. To allow for more variation in the output text, more templates can be implemented to express the same information in different ways, which are selected randomly.



CONDITIONS:
TOPIC goalscoring
time ← express:[currentevent.time, currentmatch, c]
player ← express:[currentevent.player, currentmatch, c] / nom
playergen ← express:[currentevent.player,currentmatch,c] / gen

Figure 3: Syntactic template for the sentence *Dertien minuten later liet Kluivert zijn tweede doelpunt aantekenen*

In the last stage of text generation, the *Prosody* module computes the accents and prosodic boundaries taking the properties of the Context State into account. The accentuation algorithm is based on a version of Focus-Accent Theory (van Deemter (1994); Dirksen (1992)), where binary branching metrical trees are used to represent the relative prominence of nodes with respect to pitch accent. After accentuation, phrase boundaries are assigned. The output of the LGM is an enriched text i.e., a coherent text with prosodic markers (see Figure 2), which is passed on to the SGM. The prosodic markers will be discussed in Section 3.1. For a more extensive explanation of the LGM see (Klabbers, Odijk, de Pijper, and Theune 1996).

2 Speech output generation methods

In commercial data-to-speech systems, it is important that the voice output interface be of high quality. There are several methods to provide a system with speech output, each with their advantages and disadvantages. Three methods are distinguished here, viz. the use of prerecorded speech, speech synthesis and speech concatenation.

2.1 The use of prerecorded speech

A maximum degree of naturalness can be achieved by playing back digitally stored natural speech. In the past, several information announcement systems have been created to provide such services as weather, motoring and tourist information, recipes, and bed-time stories. The speech output was created by simply making recordings of the whole information base and playing a loop or disc continuously throughout the day (Waterworth 1984). This approach has two main disadvantages. Firstly, memory and storage limitations will become a problem once the vocabulary of the system becomes too large. Secondly, the approach is highly inflexible in that entire messages have to be re-recorded to update the vocabulary.

For GoalGetter, the vocabulary consists of a limited set of carrier sentences and a more extensive set of variable words that can be inserted in the slots (*slot fillers*). Even though the vocabulary is within limits (approx. 2000 words), the total number of combinations is almost innumerable. Adding a new football player to the vocabulary would necessitate the recording of a large set of new sentences in which this player can occur. Therefore, for GoalGetter, using prerecorded speech is not a feasible method.

2.2 Speech synthesis

An alternative that yields a maximum degree of flexibility is the use of synthetic speech. This method requires much less memory than stored-waveform techniques. One way of producing synthetic speech is by *allophone* or *formant synthesis* which attempts to approximate the acoustic output of a speaker. In the DYD system the DECTALK formant synthesizer was used (Allen, Hunnicutt, and Klatt (1987) discusses its predecessor MITalk). It models the vocal tract transfer function by simulating formant frequencies, bandwidths and amplitudes. The process is controlled by 20 - 40 parameters which are updated every 5 - 10 ms. For this approach, extensive knowledge is needed on how the acoustic properties of the speech signal evolve over time. The parameters are highly correlated with production and propagation of sound in the oral tract. Various sorts of voices can be generated, as well as different speaking styles, speaking rates, etc. One of the drawbacks of this approach is that the automatic technique of specifying parameters is still unsatisfactory. The majority of parameters has to be optimized manually.

Current speech synthesizers usually produce speech by means of *diphone synthesis*. A diphone database consists of small segments excised from human speech, that cover the transitions between any two sounds of a given language. The manual preparation of the appropriate speech segments can be time-consuming, but once the inventory is constructed, there is only moderate computational power needed. Diphone concatenation is less flexible than formant synthesis, since only one voice can be synthesized. When a different voice is needed, a new diphone database has to be constructed.

Intelligibility of synthetic speech can be quite high. Diphone synthesis usually has a higher intelligibility rate than formant synthesis. However, recent evaluations show that when both types of synthetic speech are sent through a telephone channel, intelligibility decreases significantly. In GSM conditions, intelligibility drops even further (Rietveld, Kerkhoff, Emons, Meijer, Sanderman, and Sluijter 1997). Furthermore, naturalness still leaves a great deal to be desired. This leads to the conclusion that speech synthesis is not yet suitable for use in commercial applications.

Diphone synthesis has been implemented as one of the output modes in GoalGetter in order to test the prosody rules in the LGM. Because the LGM generates an orthographic representation with a unique phonetic representation[1], it is possible to do errorless grapheme-to-phoneme conversion by lexical lookup instead of rules. The phonetics-to-speech system SPENGI (SPeech synthesis ENGIne), developed at IPO, provides GoalGetter with PSOLA-based diphones (Pitch Synchronous Overlap and Add, Charpentier and Moulines (1989)). However, the prosodic and durational realization rules in SPENGI have not been optimized for the GoalGetter domain. In the rest of this paper, we focus on another output mode, namely that of speech concatenation.

2.3 Speech concatenation

The key to generating high quality speech output is to find a balance in the tradeoff between naturalness and flexibility. In that respect, concatenating prerecorded units like words and phrases appears to be a good alternative. With this approach, a large number of utterances can be pronounced on the basis of a limited number of prerecorded words and phrases, saving memory space and increasing flexibility. This technique is practical only if the application domain is limited and remains rather stable. Speech concatenation is used in most voice response services, but often the method is so straightforward, that it is not even mentioned in publications. The necessary words and phrases are simply recorded and the concatenated sentences are played back when required. This approach has two major problems:

---

[1] It could also generate a phonetic representation directly.

1. Very careful control of the recordings is needed. Usually, this is not accounted for, so that differences in loudness, rhythm and pitch patterns occur, leading to disfluencies in the speech. Phrases seem to overlap in time, creating the impression that several speakers are talking at the same time, at different locations in the room. These prosodic imperfections are often disguised by inserting pauses, which are clearly audible and make the speech sound less natural. As far as the differences in loudness are concerned, these can be remedied by manipulating the overall energy of the material after recording without loss in quality. Differences in rhythm and pitch patterns are more difficult to correct. PSOLA manipulation only works for some voices without deterioration of the speech quality.

2. The words that serve as slot fillers are recorded in one prosodically neutral version only. This makes it practically impossible to exploit the two most important features of intonation:

    (a) Highlighting *informational* structure by means of accentuation, i.e. by accenting important and new information, while deaccenting old or given information.

    (b) Highlighting *linguistic* structure by means of prosodic phrasing, i.e. by melodically marking certain syntactic boundaries and by using pauses at the appropriate places.

One simple application that takes the prosodic properties into account is a telephone number announcement system described in Waterworth (1983). In order to increase the naturalness of the long number strings, they are split into smaller chunks. Digits are recorded in three versions with different intonation contours. There is a *neutral form*, a *terminator*, with a falling pitch contour, and a *continuant*, with a generally rising pitch. Experiments showed that people preferred this method over the simple concatenation method.

Another application called Appeal, which is a computer-assisted language learning program, uses a more sophisticated form of word concatenation to deal with prosodic variations (de Pijper 1997). The words have been recorded embedded in carrier sentences to do justice to the fact that words are shorter and often more reduced when spoken in context. The duration and pitch of the words are adapted to the context using the PSOLA technique. This ensures a natural prosody, but the coding scheme may deteriorate the quality of the output speech to some extent.

3 Speech output generation in GoalGetter

Our approach to concatenating words and phrases requires no manipulation or coding of the recordings, so the quality of the speech is not affected at that point. A good speech output quality is obtained by recording several prosodic variants of otherwise identical phrases and words. In this way, a large number of utterances can be pronounced on the basis of a limited number of prerecorded phrases, saving memory space and increasing flexibility. This technique can be used whenever there is a carrier-and-slot situation, i.e., there is a limited number of types of utterances (carriers, templates) to be pronounced, with variable information to be inserted in fixed positions (slots) in those utterances. GoalGetter obviously fits this situation well. The carriers are the syntactic templates, and these have slots for variable information, such as match results, football team names, names of individual players, and so on.

To determine which words and phrases have to be recorded and how many different prosodic realizations are needed, a thorough analysis of the material to be generated is a necessary phase in the development of a phrase database.

3.1 Prosodic markers

As mentioned before the intonation of a sentence should serve to highlight informational and linguistic structure. In order to generate the proper pitch contour for a given sentence, one needs to integrate intonational, accentual and surface-syntactic information. The LGM has this information readily available and passes it on to the SGM in the form of prosodic markers. There are two basic types of markers: accent markers and phrase boundary markers. In GoalGetter, there are also special, application-specific, markers.

- *Accent markers:* A word can be either accented or unaccented. In the enriched text, accents are indicated with a double quote (") before the accented word. Deaccentuation rules are based on the given-new distinction (van Deemter 1994). As mentioned before, proper accentuation highlights informational structure. Deaccentuation is necessary in GoalGetter because accentuating given information leads to unnatural results and can even result in unintended interpretations. Recently, a third type of accent, viz. *contrast accent*, has been implemented in the LGM. However, the prosodic realizations associated with this type of accent have not yet been included in the SGM. Therefore, we leave this accent type out of consideration in this paper. The interested reader is referred to Theune (1996) (this volume) for a discussion on the prediction of contrastive accent in data-to-speech generation systems.

- *Phrase boundary markers:* Prosodic boundaries are indicated by slashes in the enriched text. The number of slashes (1, 2 or 3) denotes the strength of the boundary. The sentence final boundary (///) is the strongest one. Words which are clause-final or which precede a punctuation mark other than a comma are followed by a major phrase boundary (//). A minor boundary (/) precedes a comma and constituents to the left of an I', C' or maximal projection. This is a slightly modified version of a structural condition proposed by Dirksen and Quené (1993).

  In longer texts, containing more complicated constructions, one might want to distinguish more levels. Sanderman (1996) uses five levels for generating texts with more natural phrasing.

- *Special markers:* The symbols % and @ are used to trigger particular application-specific prosodic realizations not immediately related to accentuation and boundary marking. They are only used in the phrase concatenation mode. In order to use them in the diphone mode we need robust rules that specify how these special prosodic versions are realized, which are unavailable at the moment. The @-sign is used to mark the numbers reflecting the score. This is because in Dutch the score of a match is pronounced in a special way: the two accented numbers are realized with a so-called flat hat (a steep rise on the first accented word and a steep fall on the second one, with high pitch in between), which in Dutch is normally used only if there is no intervening boundary. The fact that the first accented word is lengthened and that a small pause seems appropriate, on the other hand, suggests that a boundary should be there.

  The %-sign is used to mark nouns that are followed by a noun phrase functioning as an adjunct, as in *de %verdediger de Boer kreeg een gele kaart* 'the defender de Boer got a yellow card'. The noun *de verdediger* can also occur in isolation where it has a longer duration and often receives an accent. In the case where it is marked with a %-sign, a different prosodic variant is chosen which is shorter and does not have an accent. This phenomenon seems to be general in Dutch and as such ought to be incorporated in the prosody rules.

3.2 Prosodic realization

Once the content and prosodic properties of the text is known, a phrase database can be developed, which provides the words and phrases that have to be concatenated. For the slot fillers, we chose to use six different prosodic realizations, one for each context described in terms of accentuation and phrasing attributes. Stylizations of these prosodic realizations are depicted in Figure 4. The special markers are not indicated in Figure 4, because they apply to a small group of words only.

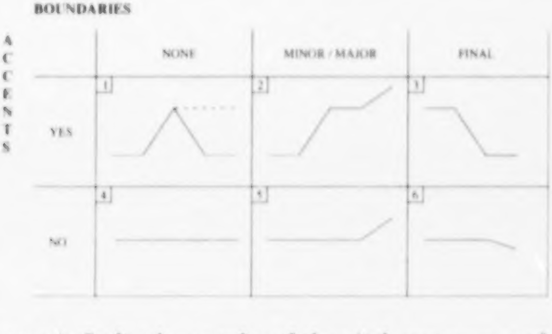


Figure 4: Stylized examples of the pitch contours needed

The six different prosodic realizations, described in terms of the IPO Grammar of Intonation ('t Hart et al. 1990), are:

1. A slot filler that is accented and does not occur before a phrase boundary is produced with the pitch movement that is most frequently used, the so-called *hat pattern*, which consists of an accent-lending rise and fall on the same syllable. This contour often corresponds to the prosodically neutral version that is used in straightforward concatenation techniques. Sometimes, the penultimate and the final accent in a sentence are combined, and instead of two hat patterns, one *flat hat* is realized. In Figure 4 this contour is obtained by combining the rise of (1) with the fall of (3). GoalGetter uses this construction mainly in time expressions that occur at the end of the sentence.

2. An accented slot filler which occurs before a minor or a major phrase boundary is most often produced with a rise to mark the accent and an additional continuation rise to signal that there is a non-final phrase boundary. A short pause is added after the word.

3. An accented slot filler which occurs in final position receives a final fall. A longer pause follows the word. This contour co-occurs with a rise in a preceding word.

4. Unaccented slot fillers are pronounced in a neutral fashion without any pitch movement associated to them.

5. Unaccented slot fillers occurring before a minor or a major phrase boundary only receive a small continuation rise. This type of words does not occur very often in the GoalGetter domain, since the LGM usually puts a minor or major phrase boundary immediately after an accented constituent.

6. Unaccented slot fillers in a final position are produced with final lowering.

When recording the material for the phrase database, the slots in the carrier sentences are filled with dummy words, so that the fixed phrases to be stored in the database can be excised easily. The slot fillers such as team and player names are embedded in dummy sentences that provide the right prosodic context. The sentences are constructed in such a way as to make the speaker produce the standard prosodic realization naturally. The intonation in the fixed phrases is not very critical, so the speaker may use his own intuitions to determine how to pronounce them.

3.3 Generating speech

In order to make a text audible, the proper words and phrases have to be concatenated by an algorithm which performs a mapping between the enriched text (with accentuation and phrasing markers), and the phrases that have to be selected. The different prosodic variants are selected on the basis of the prosodic markers. The algorithm recursively looks for the largest phrases to concatenate into sentences.

At concatenation time, the slot fillers are surrounded by short pauses of 50 ms, which are hardly perceivable, but which give the speech a less hasty character. Because the slot fillers usually contain the important information, they are supposed to stand out slightly from the rest of the sentence, which is an additional reason why introducing small pauses is not disturbing.

3.4 Selection of speaker and speaking style

The choice of an appropriate speaker is essential for the success of the application. Cox and Cooper (1981) conducted a survey to find out what properties in a human's voice make it suitable for use in a telephone information system. The results showed two important factors influencing the preferences of the listeners, i.e. agreeableness and assertiveness (which is also associated to the notion of self-confidence). In their experiments, female speakers were marked up for assertiveness whereas male speakers were marked down for that quality. Because of this property, there seemed to be a slight preference to use a female speaker in telephone announcement systems.

Speaking style also constributes to the output quality of the speech. Two important factors associated to speaking style are speaking rate and pitch range. When selecting a speaker, these factors have to be taken into account. A speaker should not speak too fast, since that gives the concatenated speech a restless, nervous quality. Especially small words like function words will sound as if they have been cut off abruptly. A speaker's pitch range should not be too excessive, as disfluencies in the speech are more likely to occur.

4 Conclusion

This paper describes a method for speech generation in the GoalGetter system. It has been demonstrated that with a sophisticated phrase concatenation technique, we can obtain speech output with a very good quality. As mentioned before, this technique is only suitable when there is a stable and fairly limited application domain. Once the language generation module generates too flexible output and the slot fillers change continuously, the phrase concatenation technique will prove to be too inflexible. Therefore, we are continuing our efforts to improve the diphone synthesis technique.

References

Allen, J., M. Hunnicutt, and D. Klatt (1987). *From Text to Speech: the MITalk System*. Cambridge: Cambridge University Press.

Charpentier, F. and E. Moulines (1989). Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. In *Proceedings EUROSPEECH'89, Paris, France*, Volume 2, pp. 13–19.

Cox, A. and M. Cooper (1981). Selecting a voice for a specified task: the example of telephone announcements. *Language and Speech 24*, 233–243.

de Pijper, J. (1997). High quality message-to-speech generation in a practical application. In J. P. H. van Santen, R. W. Sproat, J. P. Olive, and J. Hirschberg (Eds.), *Progress in Speech Synthesis*, pp. 575–586. New York: Springer-Verlag.

Dirksen, A. (1992). Accenting and deaccenting: A declarative approach. In *Proceedings of COLING 1992, Nantes, France*, pp. 865–869.

Dirksen, A. and H. Quené (1993). Prosodic analysis: The next generation. In van Heuven and Pols (Eds.), *Analysis and Synthesis of Speech: Strategic Research Towards High-Quality Text-to-Speech Generation*, pp. 131–144. Berlin - New York: Mouton de Gruyter.

Klabbers, E., J. Odijk, J. de Pijper, and M. Theune (1996). GoalGetter: From Teletext to speech. In *IPO Annual Progress Report*, Volume 31, pp. 66–75.

Rietveld, T., J. Kerkhoff, M. Emons, E. Meijer, A. Sanderman, and A. Sluijter (1997). Evaluation of speech synthesis systems for Dutch in telecommunication applications in GSM and PSTN networks. To appear in Proceedings of EUROSPEECH'97, Rhodes, Greece.

Sanderman, A. (1996). *Prosodic Phrasing: production, perception, acceptability and comprehension*. Ph. D. thesis, Eindhoven University, Eindhoven.

't Hart, J., R. Collier, and A. Cohen (1990). *A Perceptual Study of Intonation: an Experimental Phonetic Approach to Speech Melody*. Cambridge: Cambridge University Press.

Theune, M. (1996). Goalgetter: Predicting contrastive accent in data-to-speech generation. In J. Landsbergen, J. Odijk, K. van Deemter, and G. Veldhuijzen van Zanten (Eds.), *Proceedings CLIN VII*, Eindhoven.

van Deemter, K. (1994). What's new? A semantic perspective on sentence accent. *Journal of Semantics 11*, 1–31.

van Deemter, K., J. Landsbergen, R. Leermakers, and J. Odijk (1994). Generation of spoken monologues by means of templates. In *Proceedings of TWLT 8*, Twente, pp. 87–96. Twente University.

Waterworth, J. (1983). Effect of intonation form and pause durations of automatic telephone number announcements on subjective preference and memory performance. *Applied Ergonomics 14*(1), 39–42.

Waterworth, J. (1984). Interaction with machines by voice: a telecommunications perspective. *Behaviour and Information Technology 3*(2), 163–177.

Possessive affixes and complement composition

Dimitra Kolliakou *†

Abstract

A long-standing issue in the literature on clitics, namely, whether they can be best analysed as affixes or syntactically autonomous words (postlexical clitics), is here addressed with respect to the Modern Greek 'weak form' possessive pronoun. It is argued that distributional and phonological evidence strongly support an affixal analysis. Apparent difficulties for extending to possessive affixes an HPSG account that has been previously employed for pronominal affixation in Romance VP are overcome, and a *composition* approach is proposed - one which takes a categorial grammar approach to adjectives and treats them as heads in NP, and does not require flat NP structures that lack independent motivation in Modern Greek.

1 Introduction

A long-standing problem in the literature on clitics has been the issue of whether they can be best analysed as affixes or as syntactically autonomous words. According to one proposal, cf. e.g. Anderson (1992), all 'clitics' are phrasal affixes and should not be assigned the status of nodes in syntactic markers at all. Anderson discusses data from a number of languages and shows that there are very substantial similarities between the principles governing the placement of 'clitics' and those for the placement of affixes. An affixal approach to clitics will permit such generalizations to be expressed, and, moreover, dispense with, in his view, *ad hoc* syntactic categories such as *clitic* or *particle*. According to an alternative view, cf. e.g. Zwicky and Pullum (1983) and for a recent discussion Halpern (1995), it is essential to distinguish between (a) affixal clitics that are lexically attached, and (b) postlexical clitics (PLC) that have the syntax of phrases but prosodically are part of a *Clitic Group* - a prosodic category that is bigger than the *phonological word* and smaller than the *phonological phrase*, and which consists of a phonological word (host) and one or more clitics, cf. Nespor and Vogel (1986). If the view represented by Anderson is right, then what remains to be done is to work out the specifics for each particular family of affixes. Otherwise, the issue of affix versus

---

*University of Groningen, University of Newcastle

†Many thanks to Dora Alexopoulou, Jonathan Ginzburg, Aaron Halpern, Jack Hoeksema, Ineke Mennen, John Nerbonne, Ivan Sag, Ann Taylor, two anonymous reviewers, and in particular Gosse Bouma, for comments, discussion and encouragement. All remaining errors are my own. The research reported here was supported by EU TMR grant No. ERB4001GT950989.

PLC status is a burning one. I start therefore by considering this issue with respect to the Modern Greek 'weak form' possessive pronoun, henceforth, MG POSS (Sections 2 and 3). Given that I conclude that MG POSS *is* an affix, I do provide a detailed account of its morphosyntax. Thus, in Section 4 I show how a composition approach, in the spirit of previous work on pronominal affixation couched in the framework of Head-driven Phrase Structure Grammar (HPSG), can be extended to MG POSS, despite apparent empirical and conceptual difficulties for extending to NP an approach originally intended for the placement of pronominal affixes in Romance VP.

2 Basic data and previous approaches

By way of introduction, I mention a couple of essential and undisputed facts about MG POSS. First, MG POSS is an enclitic or suffix, as can be demonstrated by evidence from stress. It has often been observed that in MG lexical stress is allowed on any one of the last three syllables of a word but no further to the left (Stress Well-Formedness Condition (SWFC)). If a potential host for POSS is stressed on the antepenultimate, as e.g. kalíteros ('best'), once POSS is attached, a stress is added two syllables to the right, as in kaliteros in (1), to satisfy SWFC; the additional stress is in fact perceived as the main stress of the word, whereas the original lexical stress weakens, cf. Arvaniti (1992).

(1) o kalíterós MU filos
the best POSS.1sg friend
'my best friend'

Second, POSS exhibits a 'floating' distribution: it can attach to a specifier (2a), any prenominal adjective (2b-c), or the noun (2d). However, multiple possessive marking, as e.g. in (2e) is not allowed.

(2) a. ola TUS ta-prosfata epistimonika arthra
all POSS.3pl the-recent scientific papers
'all their recent scientific papers'
b. ola ta-prosfata TUS epistimonika arthra
c. ola ta-prosfata epistimonika TUS arthra
d. ola ta-prosfata epistimonika arthra TUS
e. *ola TUS ta-prosfata epistimonika arthra TUS

Previous approaches treat MG POSS as an affix. Sadock (1991) proposes that it is a *penultimate word suffix*. Nonetheless, as pointed out by Halpern (1995), this proposal cannot account for the patterns illustrated in (2a&b). In a similar spirit, Stavrou and Horrocks (1990) argue that MG POSS is a sister of N^0 that in D-structure attaches to N^0 or Adj^0 by means of 'a morphological rule that can apply at the syntactic level', and which yields lexical (X^0) categories. However, unlike Sadock's autolexical syntax approach, Stavrou and Horrocks make no explicit proposal concerning the interface between syntax and morphology. To the best of my knowledge, no PLC analysis of MG POSS has been previously proposed. However, a prosodic approach to the Ancient Greek possessive enclitic (AG POSS) which assigns it PLC status is provided by Taylor (1996). I summarize this proposal and consider whether it can be extended to MG.

The following data (from texts dating back to 0-300 AD) demonstrate that AG POSS can appear in two positions: (a) the NP-initial position, henceforth, **1W** (first-word-position), as shown in (3a&b), and (b) in second-position, in fact, the position after the first word (position **2W** a la Halpern (1995)), as shown in (3c&d).

(3) a. kai peisthēsontai [$_{NP}$ SOU tais-rhēmasin]
and they-will-trust [$_{NP}$ POSS.2sg the-D(AT) words-D(AT)]
'and they will trust your words'

b. ean [$_{NP}$ MOU tēn-entolēn] phulaksēi
if [$_{NP}$ POSS.1sg the-A(CC) commandment-A(CC)] he-keeps
'if he keeps my commandment'

c. aph' hēs [$_{V}$ elalēsas] autois [$_{NP}$ tas-entolas MOU]
from when you-told them-D [$_{NP}$ the-A commandments-A POSS.1sg]
'from the time when you told them my commandments'

d. metelabon hoti bareōs douleuete [$_{NP}$ tēn-kurian HĒMŌN mēteran]
I-understand that grudgingly you-serve [$_{NP}$ the-A lady-A POSS.1pl mother-A]
'I understand that you serve our lady mother grudgingly'

Following a tradition which assumes that **1W** and **2W** are related and derives the latter from the former in a 'post-syntactic' component of the grammar, Taylor provides a prosodic account of the **1W/2W** alternation: AG POSS is taken to be a *simple clitic* in the sense of Zwicky (1985), which, however, is sensitive to phonological phrase (Φ) boundaries. As shown in (4a-b) below, AG POSS is syntactically left-adjoined to NP and prosodically attaches to a linearly preceding host outside its domain, due to its enclitic status, thus giving rise to syntax-phonology mismatches of the type discussed in Klavans (1985). In case a non-optional Φ boundary ('#') precedes the clitic, i.e. one that cannot be eliminated by Φ *restructuring* in the sense of Nespor and Vogel (1986), as e.g. in case of (4c-d), *Prosodic Inversion* is triggered (cf. Halpern (1995), Taylor (1996)), which essentially amounts to allowing the linear order of host and clitic inside the *Clitic Group* to be the reverse of their linear order in the syntax. ('=' marks prosodic attachment.)

(4) a. [$_{VP}$ [$_{V}$ peisthēsontai] [$_{NP}$ [$_{CL}$ SOU] [$_{NP}$ tais-rhēmasin]]]

b. ($_{\Phi}$ peisthēsontai =SOU tais-rhēmasin)

c. [$_{VP}$ [$_{V}$ elalēsas] [$_{NP}$ autois] [$_{NP}$ [$_{CL}$ MOU] [$_{NP}$ tas-entolas]]]

d. (($_{\Phi}$ elalēsas) ($_{\Phi}$ autois)# ($_{\Phi}$ tas-entolas =MOU))

A crucial difference between AG and MG POSS is that the latter never attaches to a host outside its syntactic domain: (3a&b) are not available options in MG. To start with, this is a distributional pattern that pertains to affixes but not PLCs, as is argued at length in Halpern (1995). In addition, unless it could be shown that Φ boundaries are obligatory in MG in all contexts where Taylor takes them to be optional for AG, this distribution requires one to assume that the syntactic domain of MG POSS is never a maximal projection (NP), but rather the N'.[1] This would nonetheless go contrary to most of the literature which exclusively allows *maximal* categories to constitute the syntactic domain or scope of clitics. Let us for the moment ignore these two initial difficulties and assume that MG POSS is syntactically left-adjoined to N′ and prosodically attaches to a linearly preceding host, due to its enclitic status. Such a proposal will account for examples (2a-c), which conform to the pattern [X [$_{N'}$ POSS [$_{N'}$ Y Z]]], i.e. they contain a clitic in N′-initial position.[2] Moreover, the requirement for left as opposed to right adjunction will allow us to account for an additional fact, illustrated in (5), namely, that postnominal adjectives cannot host POSS:

---

[1] Besides, unless we take the N' as the domain, we cannot account for examples such as (2c) which are clearly not instances of second-position in NP, but rather instances of second-position (**2W**) in N'.

(5) a. ena arthro TU prosfato
one paper POSS.3.masc/neut.sg recent ('a recent paper of his')

b. *ena arthro prosfato TU

To account for **2W**, e.g. (2d), which under Taylor's proposal would be derived from a syntactic structure [$_{N'}$ epistimonika [$_{N'}$ TUS [$_{N'}$ arthra]]] by applying prosodic inversion, we must further assume that words can constitute (optional) phonological boundaries in MG NP – an assumption that has not previously been made. It can perhaps be argued that there are alternative ways for accounting for **1W/2W** alternations, e.g. by assuming that (2a-d) are all instances of discontinuous constituency, where a possessive clitic syntactically combines with an NP or N′, is permitted to interleave with its daughters, and appears after the leftmost daughter in NP or N′ – a specifier, adjective, or the noun. In the next section, I argue that the PLC approach to MG POSS, no matter whether prosodic or syntactic, will encounter a number of distributional problems.

3 Arguing for an affixal approach

A first problem for an analysis of MG POSS as a postlexical clitic is the existence of unexpected gaps in **1W** and **2W**. Along with the grammatical (6a), a prosodic or syntactic analysis along the lines of the one sketched above will allow for the ill-formed (6b), with MG POSS in **1W**, following the complement of a preceding adjective. (6b) cannot be excluded by appealing to an obligatory phonological phrase boundary to the left of tu since it is generally agreed that material inside the NP that precedes the noun plus the noun itself form a single phonological phrase.[3]

(6) a. ta [$_{AP}$ gnosta [se olus]] kolpa TU
the familiar to everybody tricks POSS.3.m/n.sg
('his tricks familiar to everybody')

b. *ta [$_{AP}$ gnosta [se olus]] TU kolpa

An approach a la Taylor would also permit the ill-formed (7b) to be derived by prosodic inversion from (7a), where POSS is syntactically left-adjoined to the N′ whose first word is the adverbial entelos. Though a syntactic PLC approach could perhaps rule out (7b) by constraining MG POSS to exclusively interleave with the daughters of an N′/NP, rather than the daughters of an AP embedded inside that N′/NP, this latter type of approach would fail to allow for the grammatical (7c). In (7c), **tu** is again embedded inside an AP, but this time it is attached to the adjective, whereas the modifier occurs in post-head position. The contrast between (7b&c) shows that APs cannot be homogeneously treated as syntactic boundaries for the purposes of clitic placement.

---

[2] In (2a), ola is taken to be the specifier, whereas ta is part of the N. This is consistent with the fact that the MG definite article does not shift the type of the phrase it occurs in from N′ to NP, rather it is a proclitic/prefix that can multiply occur in the same NP, as in (15) below.

[3] Neither could it be argued that MG POSS for some reason requires a strictly lexical host and resists a preceding phrase: not only does this assumption violate the spirit of the PLC approach, but it is also empirically refuted by the grammaticality of, say, (ta) [$_{AP}$ para poli gnosta] TU kolpa (lit.: the very much familiar POSS.3.masc.sg tricks; 'his very familiar tricks), where tu is again preceded by an AP which this time consists of an adjective and a pre-adjectival degree modifier.

(7) a. **orismenes TU [$_{N'}$ [$_{AP}$ entelos katestramenes] tixografies]**
certain POSS.3.m/n.sg totally ruined frescos ('some of its totally ruined frescos')

b. ***orismenes [$_{AP}$ entelos TU katestramenes] tixografies**

c. **[$_{AP}$ to apagorevmeno TU dia nomu] vivlio**
the forbidden POSS.3m.sg by law book ('his forbidden by law book')

The distribution patterns in (6) and (7) (as well as the example in footnote (3)) can be straightforwardly accounted for if MG POSS is viewed as part of the morphology of nominal categories (determiners e.g. **orismenes**, adjectives e.g. **apagorevmeno**, and nouns e.g. **kolpa**). In terms of Miller (1992) and Halpern (1995), this amounts to treating POSS as 'extended' inflection that exhibits *head percolation*, that is, if it is to be located inside an AP daughter of N′, it will be attached to the adjective head, if it is to be located inside the specifier, it will be attached to its determiner head, and so on; therefore, it cannot be found in the right or left edge of a daughter of N′/NP, as in the ungrammatical (6b) and (7b), respectively.[4]

A second problem for the PLC approach is the fixed order of POSS and NP-internal demonstratives – the latter must always follow the former inside NP, as shown in (8). It is commonly assumed that NP-internal demonstratives in MG have the syntactic status of adjectives. Given this, the PLC approach would predict that both (8a&b) should be grammatical, in either case **tu** being syntactically left-adjoined to an N′, and prosodically attached to a preceding adjective. An affixal approach on the other hand can circumvent this problem. Affixation is a lexical matter, therefore, there can be exceptions in a given morphological paradigm. NP-internal demonstratives can be treated as such an exception in that, unlike other adjectives, they do not participate in the morphological process of possessive affixation. Further such exceptions will be provided below.

(8) a. **ta-kenuria TU afta vivlia**
the-new POSS.3sg these books ('these new books of his')

b. ***ta-kenuria afta TU vivlia**
the-new these POSS.3sg books

A third problem for a PLC analysis is a type of obligatory 'possessive doubling' that mysteriously applies only in case of first and second person, but not in third. As illustrated in (9) below, a first person singular phrasal possessive can occur in an NP where a possessive affix is also present, but is not licit otherwise. Note also that despite the fact that phrasal possessives in MG are NPs in genitive case, when in first or second person singular and plural, they occur in accusative, since the first and second person paradigm is defective and no genitive forms are available synchronically. It is unclear how a PLC account can relate the adjoined-to-N′ POSS with a phrasal possessive that presumably occupies the noun's complement position, and in fact, in such a way that only first and second person elements are affected. In Section 4, I propose a way for accounting for such idiosyncratic doubling under an affixal approach to POSS based on composition.

---

[4] A potential counterexample for the 'head percolation' generalization is the fact that POSS cannot intervene between an adjective and its phrasal complement, e.g. the string ***ta gnosta TU se olous kolpa** is ill-formed – compare with the grammatical (6a). Notice, however, that this example would also be a problem for the prosodic inversion approach discussed above, and, moreover, for a syntactic approach assuming discontinuous constituency and which would account for the grammatical (7c). Both types of account would allow for the 'surface' ordering [Y [$_{AP}$ Adj POSS XP] Z ...]. The composition-based proposal in Section 4 provides a tentative solution to this problem by assigning possessive morphology to adjectives which do not subcategorize for a thematic complement.

(9) a. *to vivlio emena
the book mine-ACC (putatively: 'my book')

b. to vivlio MU (emena)
the book POSS.1sg mine-ACC ('my book')

Consider now some phonological evidence in favour of an affixal approach. In MG, two phonological rules can be identified whose domain of application is the word, where 'word' is to be interpreted either as a plain inflectional form, or as an inflectional form plus possessive suffix combination. First, SWFC (see above) affects lexical stress in an entirely predictable way (a) in case of inflectional affixation, where the main stress moves one syllable to the right, as e.g. in **máthima** (lesson-NOM/ACC.SG) → mathímatos (lesson-GEN.SG), and (b) in what has been traditionally referred to as *cliticization*, here analysed as 'extended' or phrasal affixation. Though in (a) and (b) lexical stress is affected in two different ways – recall that in case of possessive affixation a new stress (main stress) is added two syllables to the right of the original lexical stress, whereas the latter weakens – this difference can be accommodated in the account proposed below which identifies two types of morphology for a given nominal word: *plain morphology*, which corresponds to inflected forms, and *clitic morphology*, which corresponds to inflected forms that also bear a possessive suffix, 'clitic' bearing no theoretical significance in this piece of terminology. The second rule is the voicing of a stop when it is preceded by a nasal. Stop Voicing (SV) applies inside a plain morphology word, as e.g. in case of *αντίθεση* ('antithesis') → [andithesi], and, moreover, inside a clitic morphology word, as e.g. in *καθηγητών του* (professors POSS-3.masc.sg; 'his professors') → [kathigiton du], but not across words.

A further piece of evidence in favour of an affixal analysis is the existence of certain exceptions or 'arbitrary gaps' in the set of 'host'-POSS combinations (cf. Zwicky and Pullum (1983)). For example, though POSS can occur inside indefinite NPs, as has been previously demonstrated in (5a), particular members of the determiner class appear to resist a possessive suffix. My consultants agree that there is a contrast between (10a&b), despite that fact that **merikes** and **orismenes** do not appear to have different properties otherwise, e.g. they can both occur in the partitive construction (**merikes/orismenes apo tis fotografies** TU; 'some/certain of his pictures'), and neither of the two is licit inside definite NPs, unlike e.g. the cardinals (**i-tris/*i-merikes/*i-orismenes fotografies**; 'the three/*the-some/*the-certain pictures').[5]

---

[5] POSS can function as 'object of comparison' for a number of comparative adjectives, which are not discussed here, due to space limitations. There too, the potential for possessive affixation is lexically determined given arbitrary contrasts such as **megaliteros ap'afton** / **megaliteros** TU ('older than him' / 'older POSS.3.masc.sg'), versus **spudeoteros ap'afton** / ***spudeoteros** TU (more

(10) a. ??merikes TU fotografies
some POSS.3.m/n.sg pictures (putatively: 'some of his pictures')

b. orismenes TU fotografies
certain POSS.3.m/n.sg pictures ('certain of his pictures')

Finally, a few words about the coordination diagnostic are in order. Potential for wide scope over a coordination of hosts is taken to support a PLC approach, and vice versa, cf. Miller (1992). E.g. the ill-formedness of *Pierre les voit et écoute, as opposed to Pierre les voit et les écoute ('P. sees them and hears them') argues in favour of affix status for French VP 'clitics' such as les. Unfortunately, the same test cannot provide conclusive evidence in case of MG POSS, since the latter, unlike VP pronominal affixes, is not mandatory. Though for some speakers o-kathigitis ke i-sinaderfos MU (the-professor-MASC and the-colleague-FEM POSS.1sg) can be assigned either of the readings 'the professor and my colleague' (preferred reading) and 'my professor and my colleague', this does not unambiguously indicate that POSS can take wide scope. Rather, usage of an NP that does not contain a possessive (e.g. *o kathigitis*) can often imply the existence of a 'possessor' (in this case, student of the professor), even outside coordination contexts with the rightmost conjunct bearing POSS. Even if we were to assume that the reading 'my professor and my colleague' can only be due to the possessive suffix's taking wide scope, that still wouldn't commit us to the PLC analysis: as Miller has shown, the credibility of the coordination test varies considerably, and 'it cannot be argued that an item is necessarily *not* an affix because it can have wide scope' (1992:157). To this effect, Miller provides examples where elements for which he claims affix status appear to exhibit wide scope in coordination e.g. the definite and indefinite article in French (Le/Un collegue et ami de mon père; 'the/a friend and collegue of my father'), and shows that this is also true for elements whose affixhood is undisputed and is also reflected in the orthographic tradition, as e.g. anti in C'est un juge anti-dommage et intérêts ('He is an anti-compensation judge').

4 A complement composition approach

Previous HPSG accounts of pronominal affixation (cf. e.g. Sag and Miller (1997) and Abeillé, Godard, and Sag (1997) for French) rely on *composition*, a notion reminiscent of *division categories* in categorial grammar, originally incorporated into HPSG by Hinrichs and Nakazawa (1994). By composition, a functor (e.g. an auxiliary verb such as avoir 'have') can combine with an unsaturated argument (e.g. a participle such as donné 'given') whose valence requirements have not been satisfied, and also, *directly*, with the arguments of that participle (e.g. le livre 'the book' and à Marie 'to Mary'). Alternatively, the arguments the auxiliary 'inherits' from the participle can be realized as affixes, which allows to account for various instances of the phenomenon traditionally known as *clitic climbing*, such as e.g. in *le-lui*-avons donné ('we have given it to her/him'). This section reaches a conclusion that at first sight might appear rather surprising, namely, that an approach origin-

---

important than him / *more important POSS.3.masc.sg).

ally proposed for VP pronominal affixes in Romance can be extended to account for the possessive suffix in the Modern Greek NP.

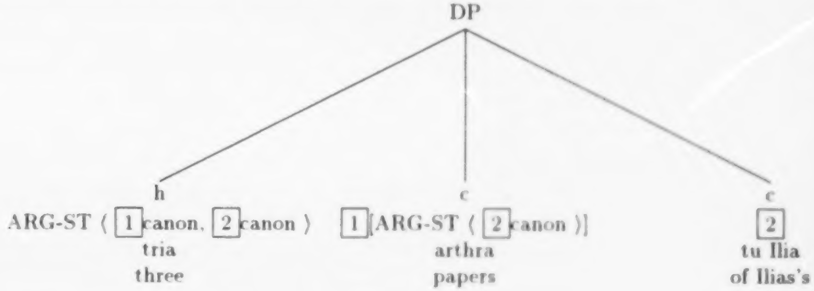
A composition approach to possessive affixation in NP requires that both determiners and adjectives should be treated as heads. In the last decade, most work on the syntax of determiners proposes their being treated as the head of the phrase they occur in. Such accounts for instance include the seminal work of Abney (1987) and within the HPSG framework the proposal of Netter (1994). Recent work in MG is also in line with this view (cf. e.g. Stavrou (1991), and Kolliakou (1995) for an HPSG approach.) Assuming a head treatment of determiners a la Netter (1994), and unlike Pollard and Sag (1994), a 'composition' determiner can be assigned the argument structure shown in (11) below by which it can select for a lexical noun and 'inherit' the arguments of that noun.[6] $\oplus$ stands for append.

(11) Determiner (preliminary version)

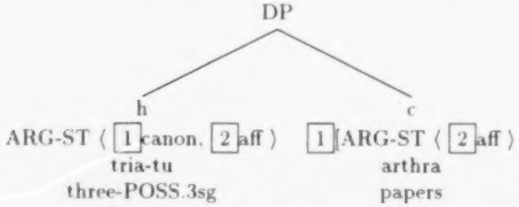
$$\left[CAT \left[\begin{array}{l} HEAD\ det \\ ARG-ST\ < N[ARG-ST\ \boxed{1}] > \oplus \boxed{1} \end{array} \right] \right]$$

The argument structure in (11) allows for two possibilities: (a) the determiner's combining with its arguments 'in the syntax', as shown in (12a), where both arguments of tria ('three') are typed *canon(ical)* – *canon* is a subtype of *synsem* and it signals that $\boxed{1}$ and $\boxed{2}$ are to be syntactically realized: signs (which are specified with a phonological value) are constrained to have canonical synsems, cf. Sag and Miller (1997), Abeillé, Godard, and Sag (1997) – and (b) the determiner's realizing the inherited possessive argument as an affix – *aff(ix)* also being a subtype of *synsem* which however is never associated with an attribute PHON, to constitute a word or phrase; this is shown in (12b) where the determiner's morphology consists of the inflected form tria ('three') and the possessive suffix tu (POSS.3.SG).

(12) a.



b.



---

[6] I am here following Sag and Godard (1994) in assuming that possessives are arguments of nouns.

However, with ARG-ST being an HPSG feature of lexical heads that does not propagate onto phrases, this type of composition approach gives rise to very flat structures like (12a) which lack independent motivation in Modern Greek. Rather, evidence from pronominalization provides support for a hierarchical structure of the type illustrated by the bracketing in (13a): the partitive tus can replace a single NP constituent that includes the noun's complement tu Ilia, as in (13b), rather than a lexical N complement of the determiner (see ill-formed (13c)).

(13) a. [tris [fili [tu Ilia]]] ('three friends of Ilias')
b. tris tus ('three of-them')
c. *tris tus tu Ilia (lit. three of-them of Ilias)

Moreover, an approach based on argument composition in the above sense will encounter a problem in case of nominal phrases which also contain adjectives: if the determiner's ARG-ST contains a lexical N and its arguments, it is not clear how adjectives can fit in. To accommodate adjectives, the argument composition analysis would require them to be treated as heads, rather than adjuncts, when combining with N′s, and AP mothers to 'dominate' noun heads and their complements. Though such a proposal is not novel in the literature (cf. e.g., in (Abney 1987)), it must be suitably formulated so as to accommodate the fact that adjectives can iterate, and, moreover, constrained so as to yield the right scoping in adverbial modification. These issues are addressed below.

Let us start with the last issue – the feasibility of a head approach to adjectives. A treatment of adjectives as heads of nominal projections is familiar from categorial grammar where adjectives are assigned the type NP/NP i.e. they are treated as functors that take an NP argument and yield another NP. In HPSG terms, this effect can be achieved (a) by modelling the types *adjective* and *noun*, which constitute appropriate values of the feature HEAD, as subtypes of a supertype *nominal*, also meant to constitute an appropriate value of HEAD, and (b) by assuming that phrases consisting of an adjective and an N′ are of type *hd-comp-ph (head-complement-phrase)*, cf. Sag (1997), or in terms of earlier HPSG work, they satisfy the head-complement ID-schema. The details of this proposal will be provided shortly.

A variety of arguments for treating adjectives and nouns in MG as partly unified categories are provided in Kolliakou (1995). I will here confine myself to some morphological and syntactic evidence in support of this position. MG adjectives and nouns fall under the same morphological paradigms and both categories are morphologically marked for case and person/number/gender agreement. (14) provides one adjective and one noun example of a given morphological paradigm in MG:

(14) a. kenuri-o (new-N(OM)/A(CC).3SG.NEUT); podilat-o (bike-N/A.3SG.NEUT)
b. kenuri-u (last-GEN.3SG.MASC/NEUT); podilat-u (bike-GEN.3SG.NEUT)
c. kenuri-a (last-N/A.3PL.NEUT); podilat-a (bike-N/C.3PL.NEUT)
d. kenuri-on (last-GEN.3PL); podilat-on (bike-GEN.3PL.NEUT)

Syntactic overlap in the distribution of APs and NPs is manifested in at least two contexts. First, both adjectives/APs and nouns/NPs can 'host' a definite article in the multiple definite marking construction, as shown in (15) below. No matter whether MG definite articles are to be treated as prefixes or proclitics (postlexical clitics), an account of their morphological or syntactic realization, respectively, needs to determine appropriate 'hosts' which they can be affixed to or cliticized onto. The supertype *nominal* allows to generalize over the MG definite article's suitable hosts.

(15) To $[_{NomP}$ (kenurio) podilato$]$ to $[_{NomP}$ kokino$]$
the (new) bike the red
'the (new) red bike'

Second, both adjectives and nouns can function as complements of higher heads in 'canonical' and 'elliptical' contexts. (16a) below shows that in MG a 'bare' singular count term can function as a 'maximal nominal projection', one which saturates a subcategorization requirement of a verb head. In the elliptical context of (16b), the same verb satisfies its valence requirement for an accusative direct object by solely combining with an adjective. Similarly, a determiner selects for a nominal complement which can be instantiated either as a noun or as an adjective, as shown in (16c&d), respectively. Modelling adjectives and nouns as subtypes of *nominal* – the category of both verb and determiner complements in MG[7] – will enable a unified account of the syntax of 'canonical' and 'elliptical' constructions to be provided – one that does not posit phonologically null noun heads in case e.g. of (16b&d), and is in the spirit of semantic approaches to ellipsis resolution that do not assume reconstruction in the syntax (e.g. Dalrymple, Shieber, and Perreira (1991)).

(16) a. Agorasa $[_{NomP}$ (kenurio) vivlio$]$
bought-1.SG (new) book
'I bought a (new) book.'

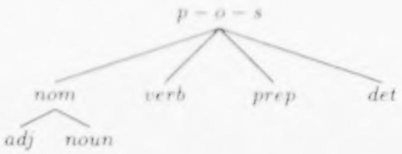
b. Exasa to vivlio mu ki agorasa $[_{NomP}$ kenurio$]$.
lost-1.SG the book POSS.1.sg and bought-1.SG new
'I lost my book and bought a new one'

c. Vrikes kanena $[_{NomP}$ isitirio?$]$
get-2.SG any ticket?
'Did you get a ticket?'

d. I times ton isitirion pikilun. Kita na vris kanena $[_{NomP}$ ftino$]$.
the prices of the tickets vary. try to get any cheap
'The prices of the tickets vary. Try to get a cheap one.'

(17) illustrates the proposed hierarchy of parts-of-speech (p-o-s) which includes *nominal*.

(17) Parts-of-speechs (p-o-s):



The feature structure in (18) below with a preliminary ARG-ST value illustrates an adjective selecting for an N′ complement, N′ being an abbreviation for objects specified HEAD *nominal* and whose valence features do not contain any canonical elements, i.e. their subcategorization requirements have already been satisfied (but see below). (18) will permit phrases containing no, one or more adjectives, the noun, and the noun's phrasal complements (if any) to syntactically combine with an adjective head which will in turn head a Nominal Phrase.[8] Case concord and agreement in person/number/gender between the adjective head and its N′ complement is straightforwardly accounted for by structure-sharing ($\boxed{1}$ and $\boxed{2}$, respectively.) The RESTR value of the adjective is specified exactly as in the adjunct approach to adjectives of Pollard and Sag (1994), by lexically unioning the set of psoas $\boxed{3}$ contributed by the N′ with the adjective's own psoa.

---

[7] Of course MG verbs also take DP complements. A slightly different proposal which accommodates this fact is provided in Kolliakou (1995).

(18) Adjective as head (with preliminary version of ARG-ST value):

$$\begin{bmatrix} SS \mid LOC & \begin{bmatrix} CAT & \begin{bmatrix} HEAD\ adj & \begin{bmatrix} PRD\ - \\ CASE\ \boxed{1} \end{bmatrix} \\ ARG-ST & < N' \begin{bmatrix} CASE\ \boxed{1} \\ INDEX\ \boxed{2} \\ RESTR\ \boxed{3} \end{bmatrix} > \end{bmatrix} \\ CONT & \begin{bmatrix} INDEX\ \boxed{2} \\ RESTR\ psoa \cup \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Note furthermore that a head approach to adjectives couched in HPSG will not encounter any particular problems with constraining adverbial scope over the adjective alone as e.g. in (19b) below, rather than over the AP, as in the incorrect (19c). HPSG does not require a one-to-one mapping between syntactic structure and scope. In fact, as shown by Kasper (1996), the 'traditional' HPSG analysis of adjectives as adjuncts of N′ provided by Pollard and Sag (1994) makes wrong predictions for recursive modification, i.e. it will incorrectly derive (19c) as the content of [[apparently indifferent] behaviour]. A treatment of adverbials as on a par with complements with respect to subcategorization, as e.g. proposed by van Noord and Bouma (1994) and Kim and Sag (1995) among many others, appears to permit a correct content value to be lexically specified, one with adverbial scope exclusively including the psoa of the adjective head, rather than a set of psoas also containing those contributed by the adjective's N′ complement. The details of such a proposal cannot be discussed here due to space limitations.

(19) a. syntax: [apparently [$_{AP}$ indifferent [behaviour]]]

---

[8] This proposal as it currently stands will allow for 'elliptical' nominals with recursive adjectives e.g. (??)**Agorases** [$_{DP}$ **kanena** [$_{NomP}$ **kenurio** [$_{NomP}$ **ftino**]]]? (lit. bought-2.SG any new cheap? putatively: '??did you buy any new cheap one?'), which are not considered to be grammatical by all speakers. One way for eliminating such nominals could be by distinguishing between adjectives and nouns, the latter being specified N+ in all instances, whereas the former being N− as a default, but N+ when combining with an NP complement – their complement's specification overriding their own default when the two are put together. Assuming that adjective heads select for N+ nominal complements, elliptical examples with recursive adjectives will thus be ruled out. A shortcoming of this proposal is however that it appeals to a notion of default which has not hitherto been commonly assumed in HPSG work, but see *order independent default unification* by Lascarides, Briscoe, Asher, and Copestake (1996). Thanks to Ivan Sag for pointing this out as an issue.

b. content: behaviour′(x) ∧ apparent′ (indifferent′(x))
c. incorrect content: apparent′(behaviour′(x) ∧ indifferent′(x))

I will now present a composition approach which (a) is similar to the one proposed for French causative constructions by Abeillé, Godard, and Sag (1997) in that the 'composed' elements are members of COMPS rather than ARG-ST, and (b) differs from the composition approaches to pronominal affixation in Romance in that it maintains hierarchical structuring by exclusively permitting affix members of COMPS to be 'inherited' by 'composition' heads.

Following Sag and Miller (1997) and Abeillé, Godard, and Sag (1997) for verbs in Romance, I assume two types of nominal words: *plain-nominal-word (pl-nom-wd)* and *clitic-nominal-word (cl-nom-wd)*. The phonology value of *pl-nom-wd* is the basic inflected form that does not bear a possessive suffix. It is computed by an inflectional function (F_M) which will have to take into account the root value specified in the lexeme type as well as information specified inside the index (person/number/gender agreement.) On the other hand, the phonology value of *cl-nom-wd* in addition incorporates a possessive suffix. It is computed by a two argument function – a simpler version of Sag & Miller's F_{PRAF} – its first argument $\boxed{1}$ being the PHON value provided by F_M, and its second argument $\boxed{2}$ the word's ARG-ST value.

(20) (a) $\begin{bmatrix} lexeme \\ PHON\ \boxed{1} \\ ARG-ST\ \boxed{2} \end{bmatrix}$ (b) $\begin{bmatrix} pl-nom-wd \\ PHON\ <\ F_M(\boxed{1})\ > \\ ARG-ST\ \boxed{2} \end{bmatrix}$

(c) $\begin{bmatrix} cl-nom-wd \\ PHON\ <\ F_{PRAF}(\boxed{1}, \boxed{2})\ > \\ ARG-ST\ \boxed{2} \end{bmatrix}$

Plain nominal forms are defined as having their argument structure list correspond to the concatenation of their valence features SPR and COMPS[9] plus a possibly empty list of gaps, in case some argument is being extracted. The list value of COMPS is unconstrained, thus allowing for plain morphology adjectives and nouns that contain an affix element in their COMPS list (and by unification in their ARG-ST list too). This is crucial for the type of composition approach proposed, one which is compatible with hierarchical NP structures. ○ stands for the shuffle operation.

(21) Valence features and ARG-ST of *pl-nom-wd*:

$$\begin{bmatrix} pl-nom-wd \\ SPR\ \boxed{0} \\ COMPS\ \boxed{1} \\ ARG-ST\ \boxed{0} \oplus \boxed{1} \bigcirc list(gap) \end{bmatrix}$$

For clitic morphology nominal forms, on the other hand, ARG-ST comprises a (potentially empty) list-of-length-one of affixes on top of the SPR and COMPS lists (and a potentially empty list of gaps.) It should be mentioned here that the only type of noun complement that can be realized in the morphology as an affix is one that is realized in the syntax as a genitive NP. A noun head in MG can take at most one phrasal genitive or affix, a fact which can be formulated as a constraint on ARG-ST lists of MG nous. An argument typed *gen*, to be realized as a phrase or affix, is also the leftmost argument inside the noun's ARG-ST. Note that the COMPS list of *cl-nom-wd* is constrained to exclusively contain canonical elements. As will become clear below, this is crucial for preventing the multiple realization of a given possessive in the same NP.

---

[9] I am following Sag and Godard (1994) in assuming that the SUBJ list of nouns is always empty and for simplicity have omitted it from ARG-ST. Notice also that since determiners are not treated here as specifiers, and, moreover, MG provides for no 'possessive determiners' e.g. in English **John's** in **John's poem**. the SPR list of most nominal examples considered in this paper is also empty. A possible exception is the case of adverbials which can perhaps be treated as specifiers of adjectives, rather than complements (see discussion above).

(22) Valence features and ARG-ST of *cl-nom-wd*:

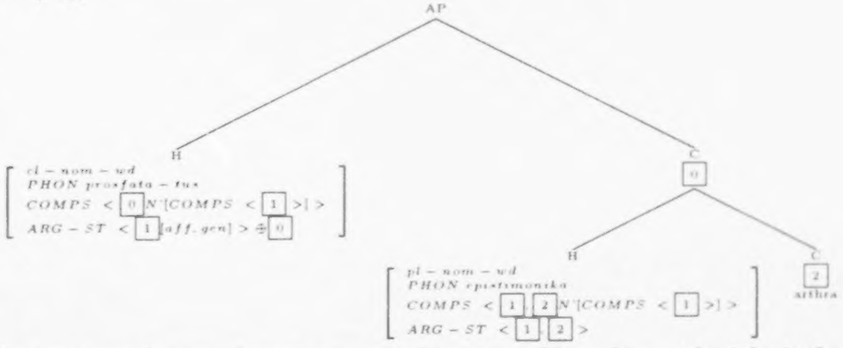
$$\begin{bmatrix} cl-nom-wd \\ SPR\ \boxed{0} \\ COMPS\ \boxed{1}(canon) \\ ARG-ST\ \boxed{0} \oplus < aff,\ gen > \oplus \boxed{1} \bigcirc list(gap) \end{bmatrix}$$

Consider now the (revised) ARG-ST list of the MG adjective:

(23) Argument structure of MG adjective (final version):

$$\left[\ ARG-ST\ < \_ > \oplus < \boxed{1}[aff,\ gen] > \oplus < N'[COMPS\ < \boxed{1} >] >\ \right]$$

The ARG-ST value in (23) is intended to replace the preliminary ARG-ST value specified in (18) above. The first slot $\langle \_ \rangle$ is reserved for the specifier. The third slot is occupied by an N′ element whose CASE, INDEX and RESTR, omitted here for simplicity, relate with the CASE, INDEX and RESTR of the adjective's exactly as was shown in (18). The crucial innovation is that N′ is specified with a potentially nonempty COMPS list which is 'inherited' by the adjective and is constrained to contain elements of type *affix* (in fact, at most one). In case the 'inherited' affix list is nonempty, two possibilities exist: (a) the affix is not present in the adjective's own COMPS list, rather it is morphologically realized, in other words, the adjective is an instantiation of *cl-nom-wd*, and (b) the affix is a member of both ARG-ST and COMPS of the adjective head and will be 'inherited' by a higher head, the adjective thus being an instantiation of *pl-nom-adj*. These two options are illustrated in the following tree-diagram, where the irrelevant for the current purposes SPR list is omitted for simplicity:
(see (2b))



The head-complement phrases represented in terms of branching nodes inherit the *head-complement-phrase* (*hd-comp-ph*) constraint:

BEST COPY AVAILABLE

(24) *hd-comp-ph* ⇒

$$\begin{bmatrix} COMPS\ \boxed{0} \\ HD-DTR\ [\ COMPS\ \boxed{0} \oplus <\ \boxed{1} \ldots \boxed{n}\ >\ ] \\ NON-HD-DTR\ <\ [SS\ \boxed{1}]\ \ldots\ [SS\ \boxed{n}]\ > \end{bmatrix}$$

(24) is in the spirit of type-specific constraints on phrasal types, as in Sag (1997). This formulation enables the first affixal element $\boxed{0}$ in the COMPS list of a given nominal to propagate to its mother's COMPS list from where it can be inherited by complement composition, the phrasal complements $\boxed{1}$ ... $\boxed{n}$ being cancelled off in the syntax. $\boxed{0}$ can alternatively be the empty list.

We can now apply the same to determiner heads and ensure that they combine with a single phrasal complement in the syntax, thus giving rise to hierarchical structures. The (revised) determiner's argument structure is as shown in (25). I also assume a plain and a clitic morphology determiner type, analogous to those provided for nominals in (20b&c) above.

(25) Argument structure of determiner (final version):

$$[\ ARG-ST\ <\boxed{1}[aff, gen]> \oplus N'[COMPS\ <\boxed{1}>]\ ]$$

Consider finally two remaining issues: (a) the requirement that post-nominal adjectives should not bear a possessive suffix, as was shown in (5) above, and (b) the obligatory 'possessive doubling' for first and second person, which was illustrated in (9). Postnominal adjectives with clitic morphology can be simply ruled out by the *Left Triggering Linear Precedence (LP) Constraint* in (26) which orders adjectives of this type to the left of their nominal syntactic sister:

(26) *Left Triggering LP Constraint:* [cl-nom-wd, adj] $\prec$ [nom]

Under the lexicalist account proposed here, 'possessive doubling' for first and second person can be accounted for by means of a lexical rule reminiscent of the one proposed by Sag and Miller (1997) for the 'floating' of **tous** ('all') from object NPs in French. The rule applies to a noun lexeme with a genitive argument in its ARG-ST list, i.e. one that can be realized as a possessive phrase or suffix. The resulting lexeme is just like the input, except that (a) the genitive argument is constrained to be of type *aff*, and, moreover, its person value specified inside the index is typed *nonthird*, a novel type subsuming first and second person, and (b) there is an additional ARG-ST member of sort *canon*, to be realized as an accusative noun phrase, that is coindexed with (and hence agreeing with) the affix. The ARG-ST list of the input is constrained to contain no accusative elements (list(nonacc)) to prevent *Possessive Doubling LR* from occurring recursively.

(27) *Possessive Doubling LR:*

$$\begin{bmatrix} lexeme \\ SS|LOC|CAT \begin{bmatrix} HEAD\ noun \\ ARG-ST\ <\ \boxed{1}[gen]\ > \oplus list(nonacc) \end{bmatrix} \end{bmatrix} \rightarrow$$

$$[\ ARG-ST\ <\ \boxed{1}[aff, nonthird]_i,\ [canon,\ acc]_i\ > \oplus list(nonacc)\ ]$$

5 Conclusion

In this paper I have presented evidence in support of an affix analysis of the MG 'weak form' possessive pronoun and discussed problems for a PLC approach to this element in terms of *prosodic inversion*. I have proposed a *complement composition* account that is reminiscent of previous HPSG work in French, but differs from such accounts in that it maintains hierarchical structuring in NP by exclusively permitting affix members of complement lists to be composed by higher heads. This approach presupposes an analysis of both determiners and adjectives in MG as heads of the phrase they occur in – a hypothesis which has been entertained in diverse theoretical frameworks (GB and Categorial Grammar), and, in addition, is independently motivated in MG.

References

Abeillé, A., D. Godard, and I. Sag (1997). Two Kinds of Composition in French Complex Predicates. *Ms.*

Abney, S. (1987). *The English Noun Phrase in its Sentential Aspect.* Cambridge, MA: MIT Press.

Anderson, S. (1992). *A-Morphous Morphology.* Cambridge: Cambridge University Press.

Arvaniti, A. (1992). Secondary Stress: Evidence from Modern Greek. In G. Docherty and R. Ladd (Eds.), *Papers in Laboratory Phonology, II*, pp. 398–419. Cambridge University Press.

Dalrymple, M., S. Shieber, and F. Perreira (1991). Ellipsis and Higher order Unification. *Linguistics and Philosophy 14:4*, 399–452.

Halpern, A. (1995). *On the Placement and Morphology of Clitics.* Stanford: CSLI Publications.

Hinrichs, E. and T. Nakazawa (1994). Linearizing Finite Aux in German Complex VPs. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-driven Phrase Structure Grammar.* Stanford: CSLI Publications.

Kasper, R. (1996). Semantics of Recursive Modification, Ms. Ohio State University.

Kim, J.-B. and I. Sag (1995). The parametric variation of English and French negation. In *Proceedings of the 14th West Coast Conference on Formal Linguistics.* Stanford: CSLI Publications.

Klavans, J. (1985). The Independence of Syntax and Phonology in Cliticization. *Language 61.1*, 95–120.

Kolliakou, D. (1995). *Definites and Possessives in Modern Greek: an HPSG syntax for noun phrases.* University of Edinburgh: PhD dissertation.

Lascarides, A., T. Briscoe, N. Asher, and A. Copestake (1996). Order Independent and Persistent Typed Default Unification. *Linguistics and Philosophy 19:1*, 1–89.

Miller, P. (1992). *Clitics and Constituents in Phrase Structure Grammar.* New York: Garland.

Nespor, M. and I. Vogel (1986). *Prosodic Phonology*. Dordrecht, The Netherlands: Foris Publications Holland.

Netter, K. (1994). Towards a Theory of Functional Heads: German nominal phrases. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-driven Phrase Structure Grammar*. Stanford: CSLI Publications.

Pollard, C. and I. Sag (1994). *Head-driven Phrase Structure Grammar*. Chicago: CSLI Publications.

Sadock, J. M. (1991). *Autolexical Syntax: a theory of parallel grammatical representations*. Chicago and London: The University of Chicago Press.

Sag, I. (1997). English Relative Clause Constructions. *Journal of Linguistics*.

Sag, I. and D. Godard (1994). Extraction of De-Phrases from the French NP. *Proceedings of NELS 24*.

Sag, I. and P. Miller (1997). French Clitic Movement Without Clitics or Movement. *Natural Language and Linguistic Theory*.

Stavrou, M. (1991). Nominal Apposition: more evidence for a DP analysis of NP. In J. Payne (Ed.), *Empirical Approaches to Language Typology*. Berlin: Mouton de Gruyter.

Stavrou, M. and G. Horrocks (1990). Klitikes ke Diktikes Antonimies mesa stin OF. In *Meletes gia tin Elliniki Glossa*. Thessaliniki: University of Thessaliniki.

Taylor, A. (1996). A Prosodic Account of Clitic Position in Ancient Greek. In A. Halpern and A. M. Zwicky (Eds.), *Second Position Clitics and Related Phenomena*. Stanford: CSLI Publications.

van Noord, G. and G. Bouma (1994). Adjuncts and the processing of lexical rules. In *Proceedings of COLING 1994*. Kyoto.

Zwicky, A. (1985). Clitics and Particles. *Language 61.2*, 238–305.

Zwicky, A. and G. Pullum (1983). Cliticization vs. Inflection: English n't. *Language 59.3*.

Presuppositions as Anaphors Revisited*

Emiel Krahmer[†]
Kees van Deemter[‡]

Abstract

Van der Sandt's theory of presuppositions-as-anaphors has been argued to be the empirically most adequate theory of presupposition projection on the market. One of the main differences between Van der Sandt's approach and its main competitor, the 'contextual satisfaction' approach, lies in the treatment of the so-called *partial match* phenomenon. In this paper, we show that the distinction between partial and full matches should be a central element of any theory of presupposition projection. However, we also argue that Van der Sandt's own formal theory, as it stands, does not offer an adequate treatment of partial matches. We then propose a modification of his formal theory, which will be argued to be more general, formally more precise, and empirically more adequate than its predecessor.

1 Introduction

Van der Sandt (1992)'s theory of presuppositions has been argued to be the empirically most successful theory on this subject available today (see e.g., Beaver 1997:983). The crux of Van der Sandt's approach is the idea that, in many respects, presuppositions behave as anaphors. A consequence of his *presuppositions-as-anaphors* view is that the notorious *projection problem for presuppositions*[1] can be reduced to the problem of resolving anaphoric pronouns. More concretely, Van der Sandt argues that presuppositions can be handled using the same mechanism which resolves anaphoric pronouns in *Discourse Representation Theory* (DRT, Kamp & Reyle 1993).

The main competitor of Van der Sandt's approach might be dubbed the *contextual satisfaction approach to presuppositions*, which has its roots in the work of Karttunen and Stalnaker, and of which Heim (1983, 1992) and Beaver (1992, 1995) are the modern (i.e., dynamic) hands on the torch. The central idea of this approach is that the presuppositions of a sentence must be entailed by the context

---

[*] The authors wish to thank David Beaver, Bart Geurts and Rob van der Sandt for comments on an earlier version of this paper, and the audiences of CLIN 96 (Eindhoven) and the Workshop on Definites (November 28th, 1996, Groningen) for comments and suggestions.

[†] IPO, Center for Research on User-System Interaction

[‡] Philips Research Laboraties, Eindhoven

[1] Langendoen & Savin (1971: 54): *"how [are] the presupposition and assertion of a complex sentence (...) related to the presupposition and assertion of the clauses it contains?"*

of interpretation in order for this context to *admit* the sentence. When Van der Sandt (1992: 349-351) compares his approach to the contextual satisfaction approach, he claims that the difference between the two approaches comes out most clearly when considering what, following Van der Sandt, might be called *the partial match phenomenon*, and of which (1) is one example.

(1) If John has an oriental girlfriend, his girlfriend won't be happy.

The possessive description *his girlfriend* triggers the presupposition that John has a girlfriend. According to Van der Sandt, this example displays a genuine ambiguity between two readings, depending on whether *his girlfriend* refers to *an oriental girlfriend* or not. The two readings may be paraphrased as (2.a) and (2.b).[2]

(2) a. If John has an$_i$ oriental girlfriend, she$_i$ won't be happy.

b. John has a$_j$ girlfriend and if he has an$_i$ oriental girlfriend (as well), she$_j$ won't be happy.

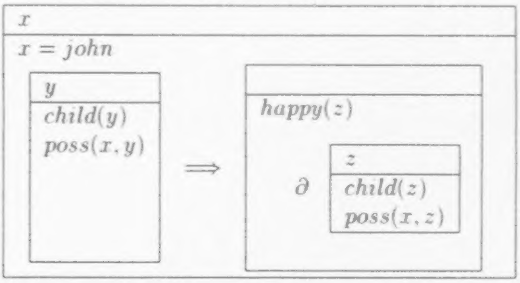
Van der Sandt claims that this is exactly what his theory predicts, while the satisfaction approach *only* gets the first reading; after all having an oriental girlfriend entails having a girlfriend.[3] However, if we apply Van der Sandt's formal theory to examples such as (1), as we will do below, we find that there is a discrepancy between his intuitions about these partial match examples and the predictions made by his formal theory. In this paper we will try to resolve this discrepancy.

2 Van der Sandt: presuppositions as anaphors

But first, let us say something about the approach to presuppositions presented in Van der Sandt (1992). Consider example sentence (3), discussed by Van der Sandt (1992:360/1) and its representation (DRS 1).

(3) If John has a child, his child is happy.

(DRS 1)



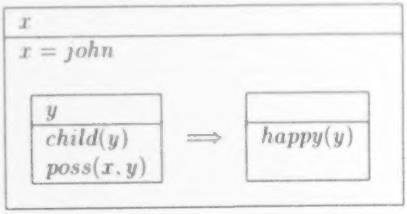
---

[2] Van der Sandt (1992:350/1) provides extra evidence for this ambiguity by showing that different continuations can eliminate one of the readings. Thus, continuing (1) with *She has always been rather jealous* (Van der Sandt 1992: 351) eliminates the (2.a) reading in favor for (2.b). Continuing (1) with *But if he has one from France, ...* will eliminate the (2.b) paraphrase.

[3] This is indeed the case for the straightforward conception of the satisfaction approach. However, Zeevat (1992:387) claims that it depends on the *representation* of the presupposition whether it is entailed or not. Zeevat does not make these ideas more precise (nor, to the best of our knowledge, does anyone else).

The consequent of the conditional contains an embedded DRS, representing the presupposition that John has a child, triggered by the possessive definite *his child*. We mark a DRS as presuppositional by prefixing it with a ∂. The ∂ operator is due to Beaver (1992), but in the present paper it is only used to syntactically distinguish presuppositional DRSs from ordinary, assertional ones. Now Van der Sandt's presupposition resolution algorithm is applied to this DRS, and starts looking for a suitable and accessible antecedent for the presupposition (as it would do for an anaphoric pronoun). Obviously, the discourse referent introduced for *a child* (i.e., y) is the ideal candidate. So, the presupposition can indeed be bound. Binding a presupposition goes as follows: the presuppositional DRS is removed from the DRS where it originates (the *source DRS*, for short), and merged with another DRS (henceforth the *target DRS*), namely the DRS which introduces the antecedent to which the presupposition is bound. Furthermore, this target DRS is extended with an equality condition which equates the referent introduced in the presuppositional DRS with the referent of the antecedent. In this way the anaphor is 'absorbed' by the antecedent (Van der Sandt 1992: 349). By binding the presupposition, (DRS 1) is transformed into (DRS 2), and this DRS can be paraphrased as *if John has a child, it is happy*.

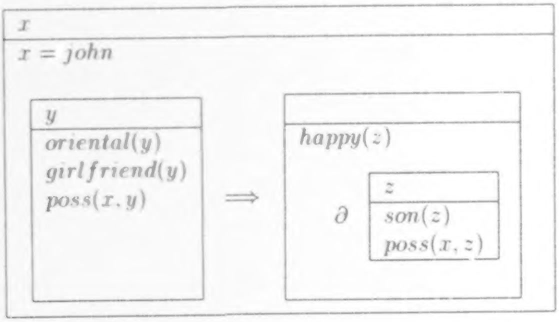
(DRS 2)



A *difference* between presuppositions and pronouns shows up when there is no suitable and accessible antecedent. In that case, a presupposition can be *accommodated*. Consider the following example with its associated DRS:

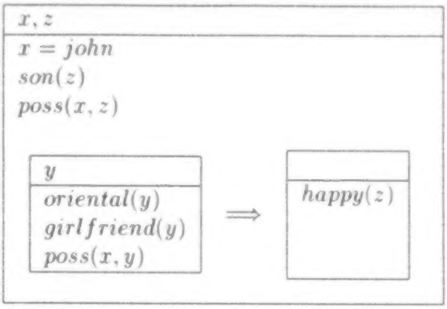
(4) If John has an oriental girlfriend, his son is happy.

(DRS 3)



Again, the resolution algorithm vill look for an accessible and suitable antecedent to bind the presupposition that John has a son. There are two accessible antecedents (John and his oriental girlfriend) but neither can qualify as suitable. Hence we *accommodate* the presuppositional DRS. If certain conditions are met,[4] accommodation takes place in the main DRS (see Van der Sandt 1992: 345 for explanation). Technically, accommodating a presuppositional DRS amounts to removing it from the source-DRS and merging it with the target DRS (which —under normal circumstances— is the main DRS). Thus:

(DRS 4)



This results in a reading which may be paraphrased as *John has a son, such that if John has an oriental girlfriend, he*$_1$ *is happy.* As this paraphrase indicates, after accommodating the presupposition the resulting DRS entails that John has a son. In general: accommodating the presupposition in the main DRS yields a 'presupposing' reading (the presupposition is projected). By contrast, from (DRS 2) it does *not* follow that John has a child; the presupposition is not projected and this produces a 'non-presupposing' reading.

It may be that there are *several* ways to resolve a presupposition. This brings us to a last, crucial ingredient of Van der Sandt's theory: the definition of a preference order over permitted interpretations. Van der Sandt defines a preference order based on the following general principles:

DEFINITION 1 (Van der Sandtian preferences)
1. Binding to a suitable antecedent is preferred over accommodation.
2. Accommodation is preferred to occur as far from the source-DRS as possible.
3. Binding is preferred to occur as near the source-DRS as possible.

In most cases, these preference rules order the set of admissible resolutions in such a way that there is *one* most preferred reading. Following Van der Sandt we will speak of a *genuine ambiguity* when there is *no* single most preferred reading. According to Van der Sandt (1992:363) partial match examples display such a genuine ambiguity, and he claims that this is one of the phenomena that his theory can account for, while the satisfaction camp cannot. However, things are somewhat more complicated. So let us now take a closer look at the partial match phenomenon.

---

[4] Of which the *Consistency* and the *Informativity constraints* are the most important ones. Roughly, the first says that accommodating a presupposition should never lead to an inconsistent DRS. Similarly, the informativity constraint states that accommodating a presupposition should never lead to a situation in which one of the sub-DRSs becomes redundant (is not informative). For more details we refer to Van der Sandt (1992: 367-369).

3 The partial match phenomenon

3.1 The empirical facts: four cases

I. Antecedent is more 'informative' than anaphor Example (1) is a prime example of this category, and we fully share Van der Sandt's intuitions that it displays a genuine ambiguity. The intuitions concerning example (1) might be a bit blurred due to a kind of lexical ambiguity in the word *girlfriend*. This is especially clear in the paraphrase of the presuppositional reading in which the globally accommodated girlfriend is John's companion in life, while the oriental girlfriend in the antecedent is more like a mistress. However, it is not difficult to find examples that do not suffer from this problem, e.g., by looking at plurals.

(5) If John has sons, his children will watch a lot of football.

This sentence displays the same kind of ambiguity as (1). Thus (5) has a presuppositional reading (paraphrasable as *John has children*$_i$, *and if he has sons, then they*$_i$ *will watch a lot of football*) and a non-presuppositional reading (*if John has sons*$_i$, *they*$_i$ *will watch a lot of football*).[5]

II. Anaphor and antecedent are 'incomparable' Consider:

(6) a. If John has sons, his young children are happy.
b. If John talks to some partygoers, the children will look at him in a strange way.

These are ambiguous in the same way as the partial match examples discussed so far. Example (6.b) is ambiguous between a presupposing reading (*there are children*$_i$ *and if John meets some partygoers, they*$_i$ *look at him in strange way*) and a non-presupposing reading (*if John talks to some partygoers, the children among them will look at him in a strange way*). (6.a) displays a similar ambiguity.

III. Anaphor and antecedent are equally 'informative' The examples in this category tend not to be genuinely ambiguous and hence they should not be categorized as partial matches. Consider:

(7) If Fido sees a cat and a mouse, he'll chase the cat and devour the mouse.

IV. Anaphor is more 'informative' than antecedent Consider (8), which is based on an example from Zeevat (1992).

(8) A man died in a car crash yesterday evening. The 26 year old man that caused the accident was found to have been drinking.

---

[5] Suppose the interpreter knows that due to some specific genetic peculiarity John and his partner can never have a *girl*. Given such background knowledge, the example (5) should not be classified in category I, but in III (anaphor and antecedent are co-extensive). This indicates that *hearer's* knowledge should be taken into account.

Examples of this kind must also be categorized as partial matches, since they constitute a genuine ambiguity. On the presuppositional reading the presupposition triggered by *the 26 year old man who caused the accident* is accommodated (i.e., the 26 year old man is still alive), and on the non-presupposing reading the presupposition is bound (i.e., he is dead).[6] Both interpretations are roughly equally plausible, as far as we can tell. However, the distribution of such examples is limited: e.g., it is difficult to find conditionals which fall in category IV. Consider:

(9) If John owns a donkey, he will be worried about the purple farmer-eating donkey on the loose. (after Beaver 1995:61)

Here, the presupposing reading seems strongly preferred over the non-presupposing one, which is at best marginal. In other words, this sentence does not seem to be ambiguous in the same way as for instance example (8) is. In Krahmer (1995:165) it is hypothesized that identity anaphora can only *add* information if the antecedent is interpreted specifically. Let us formulate this as follows.[7]

> INFORMATIVE ANAPHORS HYPOTHESIS (IAH)
> A potential antecedent with a non-specific interpretation, which is less informative than the anaphor under consideration, does not qualify as a *suitable* antecedent for the anaphor, provided that the relation between anaphor and potential antecedent is one of identity.

Thus: an (indentity) anaphor can only *add* information about its antecedent when the antecedent has a specific interpretation, and this would account for the fact that example (9) does not appear to be a genuine ambiguity. The IAH explicitly excludes non-identity anaphors, because it seems possible for such anaphors to add information about a *subset* of the antecedent.

(11) If Barney owns cows, then he will feel sorry for the mad cows.

This example indeed displays a partial match ambiguity between a non-presupposing reading (paraphrasable as *if Barney owns cows, then he will feel sorry for the mad cows he owns*) and a presupposing one (*there are mad cows$_i$, and if Barney owns cows, then he will feel sorry for them$_i$*).

Summarizing, examples of type I, II and IV display a partial match ambiguity. Of course, other factors (such as pronominal take-up in continuations or the IAH) may cause disambiguation. Similarly, intonation is an important factor which may cause disambiguation. It should be stressed however, that intonation, and in particular accenting/de-accenting, only leads to partial disambiguation. For example, de-accenting the anaphor leads to a preference for binding. When the anaphor is accented however, this will only lead to an elimination of the *identity* reading (cf. Van Deemter 1991, 1992); both the presupposing and the non-presupposing reading remain possible. Thus, when *children* in (6.b) receives a pitch accent, the reading in which all partygoers are children is excluded, but otherwise the example is still ambiguous between the presupposing and the non-presupposing reading.

---

[6] Again: extra evidence of this can be given in the form of disambiguating continuations. Continuing (8) with *The police took the drunk daredevil into custody* eliminates the non-presuppositional reading, while continuing with *This was confirmed by the pathologist who performed the post-mortem examination* eliminates the presuppositional reading.

[7] There do exist some potential counter-examples to the generalization proposed in the IAH. Consider, for example the following *'politically correct'* usage of the female pronoun:

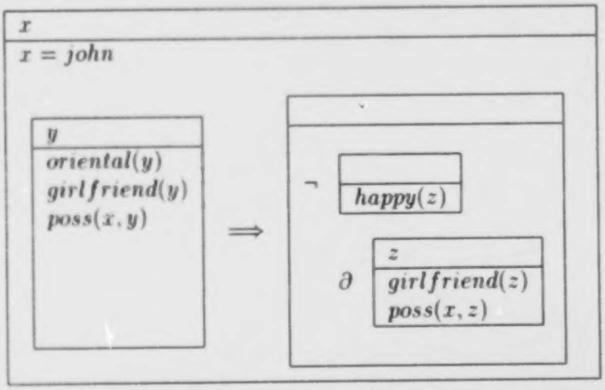
(10) If the reader has studied example (10), she might come to the conclusion that it constitutes a counterexample to the IAH.

However, we are unsure whether examples such as (10) are real counterexamples to the IAH. For instance, it has been argued by various people that pronouns are essentially devoid of semantic content (e.g., by Van der Sandt 1992), so to what extent can they *add* information?

3.2 Van der Sandt's predictions

I. antecedent is more 'informative' than anaphor Let us reconsider Van der Sandt's own (1) again, and let us construct a DRS for this example.

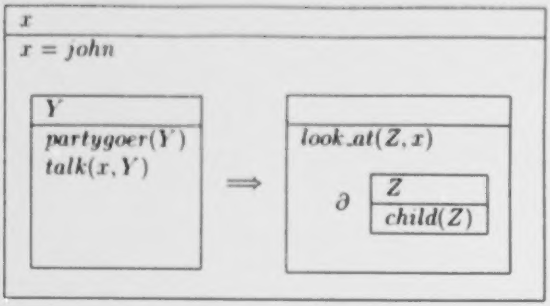
(DRS 5)



If we feed (DRS 5) to Van der Sandt's resolution algorithm, it will first start looking for a discourse referent which is accessible and which satisfies the conditions of being a girlfriend, and standing in the possessive relation with John. But such a referent is easily found: y meets all the conditions. As we saw in section 2, definition 1, binding a presupposition to a suitable antecedent is preferred over accommodating. In the DRS we are currently discussing, it seems that y is a perfectly suitable and accessible antecedent, so it is unclear how Van der Sandt (1992)'s formalism can avoid binding the presupposition, which would make the non-presupposing reading (given in (2.a)) the primary reading of (1) and hence would predict that this example is not truly ambiguous after all. It is conceivable that binding is defined in such a way that y is no longer a suitable antecedent, but then binding is precluded and accommodation is the only option. Consequently, no ambiguity between binding and accommodation is predicted either. Hence, one might say that Van der Sandt's formal theory does not fully implement the intuitions sketched in the first part of Van der Sandt (1992).

II. anaphor and antecedent are 'incomparable' The same problem applies as in category I, and other problems apply in addition. For example, consider (6.b). Here is the Van der Sandtian DRS for this example.

(DRS 6)



If we feed (DRS 6) to the algorithm, it will again look for an accessible, suitable antecedent.[8] It is unclear to us whether *some partygoers* is a suitable antecedent for *the children* according to Van der Sandt's algorithm, but it yields undesired results either way. The situation is roughly the same as for (DRS 5): either Y (the partygoers) is *not* a suitable antecedent for Z (the presupposed children). In that case, the presupposition is preferably accommodated and no genuine ambiguity results. If, by contrast, Y (the partygoers) *is* a suitable antecedent for Z (the children), binding is preferred and, as before, no ambiguity results. But in this case, there is an additional problem, which has nothing to do with preferences between interpretations. If the presupposition gets bound, it is 'absorbed by the antecedent', and this results in a reading which may be paraphrased as *if John meets some partygoing children, they'll look at him in a strange way*. This reading seems wrong. Binding should appear *in situ*, that is: the presupposition to be bound should not be merged with the target DRS, but with the source DRS.[9] Summarizing, we think that the binding reading of (6.b) should be *if John talks to some partygoers, the children among them will look at him in a strange way*. The situation in which all the children happen to be partygoers can be viewed as a special case, which is typically marked by the lack of an accent on *children* (see above). Finally, the reader may easily verify that the same problems are encountered for case IV.

---

[8] We follow the notation for plurals used by Van der Sandt (1992: 370), where he explains how an example similar to our (6.a) should be dealt with. The capitals are discourse referents standing for sets of objects. All predicates in this paper are 'strictly distributive' in the sense of Kamp & Reyle (1993, 407). E.g., $child(X)$ has the intuitive interpretation that all elements of X are children. In Kamp & Reyle (1993) this is denoted as $child^*(X)$. We will omit the $*$ superscript where this can be done without creating confusion.

[9] Consider another example:

(12) If John has children, he'll spoil the little bastards.

We are well-aware of the fact that epithets like *little bastards* have some peculiar properties. Nevertheless, they serve nicely to further illustrate the point about binding mentioned in the main text. If we bind the presupposition triggered by the definite description in Van der Sandt's way, we end up with a reading which may be paraphrased as *if John has children and they are little bastards, then he'll spoil them*. In other words: the children are only spoiled if they are little bastards. In our opinion, the right reading for this example (disregarding the differences between presupposed and asserted material) is something like *if John has children, they'll be little bastards and he'll spoil them*.

4 An alternative

In the previous section (3.1) we argued that an anaphor and an antecedent stand in a *partial match relation* if the two are *not* co-extensive. Moreover, on the partial match interpretation, a sentence is ambiguous between a presupposing and a non-presupposing reading (although we have seen that certain independent factors may cause disambiguation). In other words, we support the *intuition* sketched in Van der Sandt (1992:349-351). However, if we apply the *formal theory* (i.e., the presupposition resolution algorithm) of Van der Sandt (1992) to the partial match examples (as done in 3.2), we encounter two problems: (*i*) the algorithm does *not* generate the required genuine ambiguity in the case of a partial match, and (*ii*) not all the binding readings are correct.

We propose a modified version of Van der Sandt's resolution mechanism. One central ingredient is the use of so-called *context variables*. *Binding* will be viewed as contextually restricted quantification, where the relevant context is provided by the anaphoric antecedent. *Accommodation* will be a contextually restricted variant of the usual accommodation procedure. To arrive at all the different possible (binding or accommodation) interpretations of a given sentence containing a presupposition, we exploit Van der Sandt's resolution mechanism, with its use of unresolved representations. However, we make some modifications to the resolution mechanism as such, taking the notion of partial match into account by paying more attention to properties of potential antecedents. When antecedent and anaphor stand in a partial match relation, the algorithm will generate a real ambiguity. This entails that our modification of the algorithm yields a modified, *partial* preference order between possible interpretations.

4.1 Preliminaries

Van der Sandt (1992) is mostly based on the basic, first-order DRT fragment. The kind of examples we are interested in, and the treatment we have in mind for them, calls for two extensions of this basic DRT fragment.

Plurality and quantification in DRT In the following, we adopt the basic treatment of plurality and quantification outlined in Kamp & Reyle (1993, ch. 4). Kamp & Reyle use an algebraic 'Link-style' interpretation of plurality, in which the domains contain atomic as well as non-atomic entities. Following the convention of Kamp & Reyle (1993), we use boldface lowercaps variables ($\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, ...) to range over both individual (or *atomic*) referents *and* plural (*non-atomic*) referents. Lowercase variables ($x, y, z, \ldots$) are used for individual referents, and uppercase variables ($X, Y, Z, \ldots$) for plural referents. This convention entails that general definitions contain boldface referents, and actual examples do not.

We also adopt the treatment of generalized quantifiers in Kamp & Reyle (1993, ch. 4) in terms of *duplex conditions*. In general, a generalized quantifier (which we shall denote as DET) is a relation between two sets of (atomic) entities, say A and B, and this is represented by Kamp & Reyle as a condition consisting of two boxes A' and B', representing A and B respectively, separated by a capsized box which contains the quantifier and the variable it applies to. The quantifier gets its usual interpretation as known from *generalized quantifier theory* (GQT; for technical details on quantifiers in DRT we refer to Kamp & Reyle 1993:425-427). Here are GQT-style definitions of singular and plural *the* (d and d' atomic):

$the^{sg}(A)(B)$ is true with respect to a model M iff
$\exists d \in D : d \in A \ \& \ \forall d' \in D(d' \in A \Rightarrow d' = d) \ \& \ d \in B$

$the^{pl}(A)(B)$ is true with respect to a model M iff
$\exists d \in D : d \in A \ \& \ \forall d' \in D(d' \in A \Rightarrow d' \in B)$

It is worth pointing out that Kamp & Reyle still distinguish *indefinites* from 'truly' quantificational determiners, and we will follow this practice, as we have done so far. Concretely, this means that indefinite NPs of the form DET CN, where DET is either *a(n)*, *some* or empty (in the case of bare plurals) introduce a fresh discourse referent in the current DRS.

Context variables In Westerståhl (1985) the notion of contextually restricted quantification is introduced, motivated by examples such as the following:

(13) The children were having a lot of fun.

Clearly this is not a statement about all the children in the universe. According to Westerståhl, the definite determiner acts as a *context indicator which signals the presence of a context set C* (Westerståhl 1985:60) in such a way that *the children* denotes $C \cap child$, i.e., a contextually restricted subset of the set of all children.

In our revision of the presuppositions-as-anaphors theory, we will use context variables, which we will represent as $C, C', \ldots$. These context sets are just discourse referents (compare Westerståhl 1985:70). Below, we let every NP introduce an ordinary discourse referent and a fresh context set and our modified presupposition resolution algorithm explicitly operates on these context sets. It is worth to emphasize that the use of context sets in this paper merely facilitates the resolution process. Besides introducing contextual variables, we also employ 'contextually restricted' predicates. That is, we use conditions like $man^{C}(john)$ which have as intuitive interpretation: *john* is a *man* and an element of the context set C.[10]

4.2 The presuppositions of definite descriptions

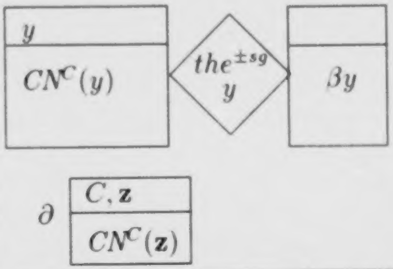
When the DRS construction algorithm encounters a definite description, the [*the* CN] rule is activated. Here *CN* is the representation of CN (in singular form, where CN is a possibly complex common noun phrase), and $\mathbf{z}$ is z or Z depending on the number of the CN. This rule is a variant of CR.NP [Quant = +], Kamp & Reyle (1993: 318, 347). Definite descriptions are generally assumed to trigger an *existence presupposition*. In this rule this is modelled as follows: a definite description presupposes that there is some context set C which has a non-empty intersection with the CN denotation.

---

[10] Formally, if η is a noun representation: $M \models_f \eta^{C}(\mathbf{x})$ iff $f(\mathbf{x}) \in I_M(\eta) \cap f(C)$. This clause is a variant of cause (ii:g:i) of definition 4.3.7 of Kamp & Reyle (1993:426).

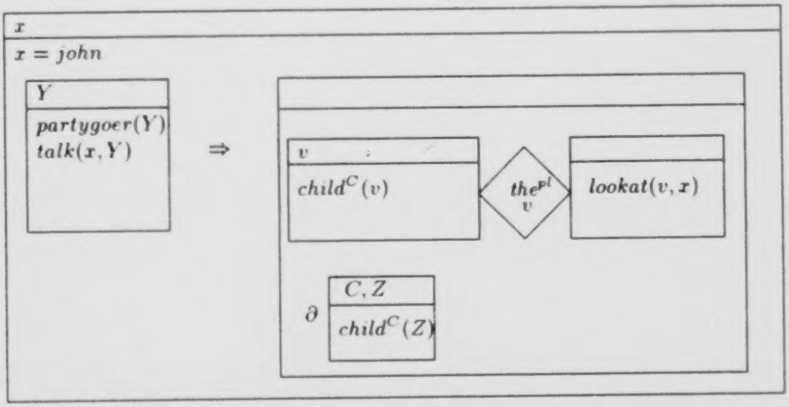
[DET CN] Rule, for DET = *the*

Upon encountering an S of the form $\alpha\beta$ or a VP of the form $\beta\alpha$, with α a definite description (of the form *the* CN[± sg]), replace S or VP with the following presuppositional DRS and duplex condition, where y and $\mathbf{z}$ are fresh discourse referents and C is a fresh context variable.



To illustrate the [*the* CN] rule, consider example (6.b) again. This sentence is represented by (DRS 7). *Some* is indefinite: it introduces a fresh (non-atomic) discourse referent Y. *The children* is handled by our definite descriptions rule: it introduces a presuppositional DRS, with the intuitive interpretation that there is some context set C which contains children, and a duplex condition, which expresses that all children in this context set C look at John in a strange way.

(DRS 7)



4.3 The modified presupposition resolution algorithm

When Van der Sandt's resolution algorithm encounters a presuppositional DRS it will first try to bind this presupposition to an antecedent, and our modified algorithm will do the same. This immediately raises a question: what qualifies as an antecedent? The answer of Van der Sandt (1992) is simple: every suitable discourse referent which is accessible from the DRS containing the presuppositional DRS is a potential antecedent. Van der Sandt (1992) does not specify what makes a referent suitable. In our opinion, the main factor in determining the suitability of a discourse referent is the phrase which lead to the introduction of the referent.

(14) a. Yesterday, an_1 uncle of mine bumped into a_2 man. The_i man fell down.

b. Yesterday, a_2 man bumped into an_1 uncle of mine. The_i man fell down.

We contend that in both (14.a) and (14.b), the definite *the man* is strongly preferred to be coindexed with *a man* (i.e., $i = 2$), even though obviously both 1 and 2 are male persons. This is due to the fact that 1 is introduced as a *man*, while 2 is introduced as an *uncle*.[11] This shows that the resolution algorithm should not only take discourse referents into account, but also properties of the phrase which lead to the introduction of the referent. In particular, we are interested in the possible values which a discourse referent can have according to the denotation of the phrase with which the referent is associated. For this purpose, we will use *value sets*. For the examples in (14) it is the CN which determines the relevant value set. But for other phrases which lead to the introduction of a referent (e.g., proper names) this may be different. Consider the indefinite description *a man with a hat*, and suppose that it triggers the introduction of a discourse referent y. Then the value set of y in a model M and with respect to an assignment f, denoted as $\mathrm{VAL}(y, [\![\, [y, z \mid man(y), hat(z), with(y, z)] \,]\!]_{M,f})$, is given by:[12]

$$\{d \in D \mid d \in I(man) \mathbin{\&} \exists d' \in D : d' \in I(hat) \mathbin{\&} \langle d, d' \rangle \in I(with)\}$$

In words: the value set of y in M is the set of men with a hat in M. Notice that in the case of atomic predicates P, the value set $\mathrm{VAL}(\mathbf{x}, [\![P(\mathbf{x})]\!])$ equals the predicate denotation $[\![P]\!]$.[13] In those cases, we will use the predicate denotation as value set. Below we will consider pairs $\langle \mathbf{x}, \mathrm{VAL}(\mathbf{x}, [\![\Upsilon]\!]) \rangle$ consisting of a discourse referent and a corresponding value set as antecedents. We are now in the position to sketch our modified resolution algorithm. The input of the algorithm is an underspecified

---

[11] Another illustration of this is the following minimal pair.

(15) If John is looking at some $[_{CN}$ children], who play basketball, then the children will strive to impress him.

(16) If John is looking at some $[_{CN}$ children that play basketball], then the children will strive to impress him.

The only difference between the two examples is that in (15) a referent is introduced by *children* while in (16) it is introduced by *children that play basketball*. Now, example (16) is ambiguous and (15) is not. The latter only has a non-presupposing reading; we cannot continue this example with *They know he is a talent scout for Utah Jazz*. Example (16), on the other hand, displays a partial match ambiguity between a presupposing and a non-presupposing reading.

[12] Reference to models and assignment functions is omitted where this can be done without creating confusion.

[13] In general: suppose that a phrase α leads to the introduction of a (atomic or non-atomic) discourse referent $\mathbf{x}$. The value set of $\mathbf{x}$ with respect to Φ (where Φ is the DRS which results from α) and given a model M and an assignment function f is defined as $\mathrm{VAL}(\mathbf{x}, [\![\Phi]\!]_{M,f}) =_{\mathrm{def}} \{d \in D \mid M \models_{f \cup \langle \mathbf{x}, d \rangle} \Phi\}$. The embedding function f is only needed when Φ is not a *proper* DRS, i.e., when some condition in Φ contains a discourse referent that is not introduced in Φ, that is, if the η phrase contains a pronoun (e.g., *the man that saw him*).

DRS containing at least one unresolved presuppositional DRS. As we have seen, for definite descriptions this presuppositional DRS will be of the form:

(DRS 8) $\partial \begin{array}{|c|}\hline C, \mathbf{y} \\ \hline CN^{C}(\mathbf{y}) \\ \hline \end{array}$

For each presuppositional DRS there is a list of Potential Antecedents (PA), and as argued above this is a list of accessible discourse referents plus their respective value sets. This list is ordered by *nearness* to the presuppositional DRS, i.e., the first element on the list is the nearest referent and the last element is the one farthest away. In general, this list will appear as follows:[14]

$$\mathrm{PA} = \langle\langle \mathbf{x}_1, \mathrm{VAL}(\mathbf{x}_1, [\![\Upsilon_1]\!])\rangle, \ldots, \langle \mathbf{x}_i, \mathrm{VAL}(\mathbf{x}_i, [\![\Upsilon_i]\!])\rangle, \ldots, \langle \mathbf{x}_n, \mathrm{VAL}(\mathbf{x}_n, [\![\Upsilon_n]\!])\rangle\rangle$$

The modified resolution algorithm is now going to try and bind the presuppositional (DRS 8), triggered by the definite description, to an element of the list of potential antecedents. We use PRES_M to denote the value set of the referent associated with the phrase which triggers the presupposition. In the case of definite descriptions (as in (DRS 8)), $\mathrm{PRES}_M = \mathrm{VAL}(\mathbf{y}, [\![\, [\mathbf{y} | CN(\mathbf{y})] \,]\!]_M)$. In general, PRES equals $\mathrm{VAL}(\mathbf{y}, [\![\Upsilon]\!])$, where Υ is the DRS representing the phrase which has led to the introduction of $\mathbf{y}$. Similarly, we use ANT^i_M as an abbreviation of $\mathrm{VAL}(\mathbf{x}_i, [\![\Upsilon_i]\!]_M)$, for some $\langle \mathbf{x}_i, \mathrm{VAL}(\mathbf{x}_i, [\![\Upsilon_i]\!]_M)\rangle \in \mathrm{PA}$.[15]

```
IF $\exists i(\mathrm{PRES}_M = \mathrm{ANT}^i_M$, in all H-models $M$)
THEN BIND
ELSE IF $\forall i(\mathrm{PRES}_M \cap \mathrm{ANT}^i_M = \emptyset$, in all H-models $M$)
        THEN ACCOMMODATE
        ELSE (Partial Match!)
                BIND OR ACCOMMODATE
```

In words: the algorithm first checks if there is a potential antecedent with the *same* denotation as the presupposition in all H-models. If it finds one, it is a *full match* and the presupposition will be bound (both the BIND and the ACCOMMODATE operation will be defined below). If the value set of the presupposition is *disjoint* with the value sets of all potential antecedents, the presupposition is accommodated. The other cases are partial matches: there is no antecedent with the same value set as the presupposition, but there *is* an antecedent which matches partially, i.e., has a non-empty intersection with the presupposition in some H-model, then the

---

[14] Nearness has an obvious formal definition in terms of subordination, see Krahmer & Van Deemter (1997). Instead of a list, PA should be a partial order (because several discourse referents may be introduced at the same level and these are 'equally far away' from the source-DRS), but we will ignore this here.

[15] It has been noted in footnote 5 that the hearer's background knowledge may cause disambiguation. This was illustrated by example (5). It was argued that if the interpreter knows that John and his partner do *not* have *daughters*, this example only has a non-presupposing reading. Therefore, our algorithm will not quantify over all possible models, but rather over all models which are in accordance with the interpreter's knowledge state. For this case, the interpreter's *H-models* (H for hearer) will not include models in which John has daughters. In what follows, specific hearer knowledge will not be taken into account, unless noted otherwise.

BEST COPY AVAILABLE

presupposition can either be accommodated or bound to this partially matching antecedent. Before we can return to our example, we have to define the notions BIND and ACCOMMODATE. To begin with the former, it follows from the algorithm that we BIND the presuppositional (DRS 8) if an antecedent $\langle \mathbf{x}_i, \mathrm{ANT}^i \rangle \in \mathrm{PA}$ has been found such that ANT^i is either coextensive with the value set PRES (full match), or has a non-empty intersection with it (partial match).

DEFINITION 2 (BIND)
$\langle \mathbf{x}_i, \mathrm{ANT}^i \rangle$ is the nearest antecedent in PA:

1. merge the presuppositional DRS with the source DRS, and
2. add a condition $C = \mathbf{x}_i$ to the source DRS

Binding is *in situ* (the presuppositional DRS is *not* moved to the target DRS, where $\mathbf{x}_i$ was introduced, as in Van der Sandt 1992). Moreover, it generalizes to non-identity anaphors since only the *context set* is equated with a set of objects, as illustrated for example (6.b) below. ACCOMMODATE is defined as follows:

DEFINITION 3 (ACCOMMODATE)
The main DRS is the (initial) target DRS:

1. remove the presuppositional DRS from the source DRS and merge it with the target DRS,
2. add a condition $C = \mathbf{D}$ to the target DRS[16]
3. check whether the result satisfies the Van der Sandt conditions, (consistency, informativity &*c*). If not, redo 1-3 with a new target DRS: the one immediately subordinated by the old target DRS

The second clause states that the context variable C is equal to the domain of discourse, thereby neutralizing the effect of C. It is worth emphasizing that this is done to keep the differences with Van der Sandt to a minimum: it entails that our ACCOMMODATE is the same operation as Van der Sandt (1992)'s accommodation.[17] Reconsider our example (6.b), and its associated (DRS 7). (DRS 7) is the input for our modified resolution algorithm. The list of potential antecedents for the presuppositional DRS looks as follows:[18] $\langle\langle Y, [\![partygoer]\!]\rangle, \langle x, \{john\}\rangle\rangle$. Let us assume that there is no specific hearer knowledge, then there will be an H-model M such that $[\![partygoer]\!]_M \neq [\![child]\!]_M$. In other words: there is no full match between *some partygoers* and *the children*. However, there will also be an H-model M in which $[\![partygoer]\!]_M \cap [\![child]\!]_M \neq \emptyset$ (after all, children can be partygoers). In other words: the algorithm predicts that this is a partial match, and a genuine ambiguity between a binding and an accommodation reading ensues. (DRS 9) results when we BIND the presuppositional DRS. This DRS can be paraphrased as *If John talks to some$_i$ partygoers, then there are children$_j$ among them$_i$, and all of*

---

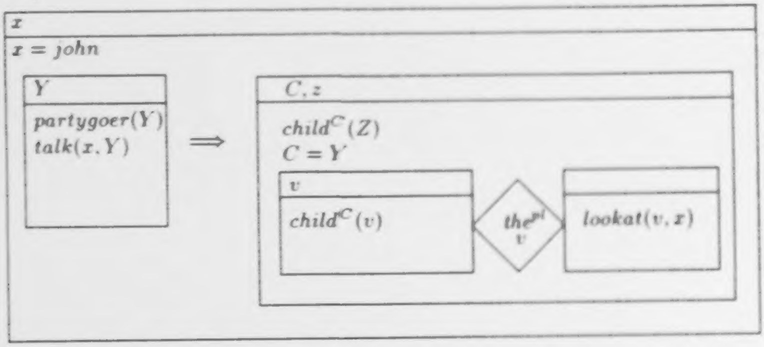
[16] The constant $\mathbf{D}$ refers to the domain of discourse: $[\![\mathbf{D}]\!]_M = I_M(\mathbf{D}) = D_M$.

[17] Krahmer & Van Deemter (1997) explore an alternative definition, where C is not necessarily equated with the entire domain, but rather with a contextually salient group of individuals.

[18] Since, $\mathrm{VAL}(Y, [\![\, [Y \mid partygoer(Y)] \,]\!])$ is equal to $[\![partygoer]\!]$, and $\mathrm{VAL}(x, [\![\, [x \mid x = john] \,]\!])$ is equal to $\{john\}$, we opt for the more simple notation.

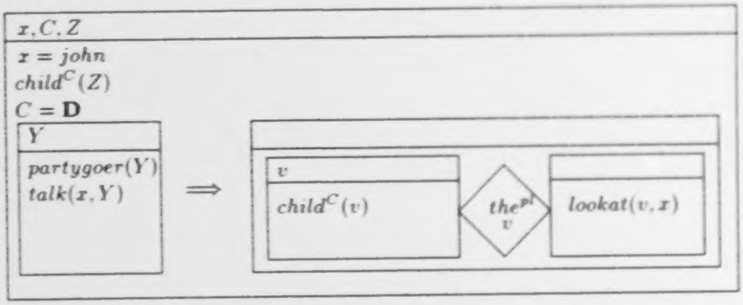
the children among the$_i$ *partygoers look at him in a strange way*. And, as argued above, this is the correct binding interpretation.

(DRS 9)



The second reading comes about via a global application of ACCOMMODATE:

(DRS 10)



In words: *There are some*$_i$ *children, and if John talks to some partygoers, all these*$_i$ *children will look at him in a strange way*.[19] Summarizing, if we feed the representation of example (6.b), (DRS 7), to the modified resolution algorithm, it decides that there is a partial match between the presupposition triggered by *the children* and its antecedent *some partygoers*. The corresponding ambiguity is between (DRS 9) and (DRS 10) for the non-presupposing/binding and presupposing/accommodation interpretation respectively.

5 Concluding remarks

We have seen (section 3.2) that the otherwise empirically successful formal theory of Van der Sandt (1992) does not always make the right predictions in cases where there is a *partial match* between a presupposition and a potential antecedent for this presupposition. We think that the problems with partial matches can be solved by refining and extending Van der Sandt's algorithm, and we have tried to do so. The resulting version of the presuppositions-as-anaphors theory differs from the one of Van der Sandt (1992) mainly in these respects: (1) It contains a precise definition of the 'partial match' phenomenon; (2) we have modified the resolution algorithm in such a way that —in accordance with Van der Sandt's intuitions— partial match sentences come out as genuine ambiguities; and (3) binding is redefined in such a way (*'in situ'*) that non-identity anaphors receive adequate interpretations. In this paper we have opted for a *frog perspective* on presupposition projection, focussing on one kind of presupposition triggers: definite descriptions. However, in Krahmer & Van Deemter (1997) it is shown that there are few impediments to extending the approach described here: a general Noun Phrase presupposition scheme is proposed, applying to *any* NP, and it is shown that the modified resolution algorithm yields the required results in these cases as well.

---

[19] Under the alternative definition of ACCOMMODATE mentioned in footnote 17 the resulting reading can be paraphrased as *there is a contextually salient group of children, and if John talks to some partygoers, all these children will look at him in a strange way*.

References

[1] Beaver, D. (1992). The Kinematics of Presupposition. In: P. Dekker & M. Stokhof (eds.) *Proceedings of the 8th Amsterdam Colloquium*, ILLC, Amsterdam, 17-36

[2] Beaver, D. (1995). *Presupposition and Assertion in Dynamic Semantics*. Ph.D. thesis, Edinburgh

[3] Beaver, D. (1997). Presupposition. In: J. Van Benthem & A. ter Meulen (eds.), *The Handbook of Logic and Language*, Elsevier, 939-1008

[4] Van Deemter, K. (1991). *On the Composition of Meaning*. PhD thesis, Amsterdam

[5] Van Deemter, K. (1992). Towards a Generalization of Anaphora. In: *Journal of Semantics* **9**: 27-51

[6] Geurts, B. (1995). *Presupposing*. PhD Dissertation, Osnabrück

[7] Heim, I. (1983). On the Projection Problem for Presuppositions. In: M. Barlows (ed.) *Proceedings WCCFL*, **2**, Stanford University, Stanford, 114-125

[8] Heim, I. (1992). Presupposition Projection and the Semantics of Attitude Verbs. In: *Journal of Semantics* **9**: 183-222

[9] Kamp, H. & U. Reyle (1993). *From Discourse to Logic*. Kluwer, Dordrecht

[10] Krahmer, E. (1995). *Discourse and Presupposition*. PhD thesis, Tilburg, `http://cwis.kub.nl/~fdl/research/ti/faces/Ek/ek.htm`

[11] Krahmer, E. & K. van Deemter (1997). Presuppositions as Anaphors: Towards a Full Understanding of Partial Matches. In: P. Dekker, J. van der Does & H. de Hoop (eds.) *Proceedings of the definites workshop*, University of Utrecht

[12] Langendoen, D. & H. Savin (1971). The Projection Problem for Presuppositions. In: C. Fillmore & D. Langendoen (eds.) *Studies in Linguistic Semantics*, Holt, New York, 55-60

[13] Van der Sandt, R. (1992). Presupposition Projection as Anaphora Resolution. In: *Journal of Semantics* **9**: 333-377

[14] Westerståhl, D. (1985). Determiners and Context Sets. In: J. van Benthem & A. ter Meulen (eds), *Generalized quantifiers in natural language*, Foris

[15] Zeevat, H. (1992). Presupposition and Accommodation in Update Semantics, *Journal of Semantics* **9**: 379-412

Modeling Coordination by Means of Operations on Strings and on Derivation Trees

Carlos Martín-Vide*
Gheorghe Păun†‡

Abstract

Several operations on strings are introduced, as models of the phenomenon of coordination in natural languages. Their relationships with other string operations is investigated. On this basis, the closure properties of families in the Chomsky hierarchy are obtained. In particular, we prove that the family of context-free languages is not closed under all but one of these operations. This special case concerns the coordination defined only between strings with a common syntactic structure (both strings have derivations described by identical trees, modulo the coordinated subwords). Some interpretations of these results are mentioned.

1 Coordination; Some Variants

The idea we start from is that in a given coordinated structure all the conjuncts are of the same type and status, and the coordination as a whole is of the same type and status as its subparts. This means that it is not possible to define only one head in the construction (coordination is not a projection of the conjunction, and there is no dependence between the two conjuncts), making it impossible to apply any general principle of hierarchical construction of the sentence (X-bar, for instance).

Coordination is basically a recursive phenomenon, because it builds phrase structures (trees associated to strings) of any length. Several studies of coordination start from the following generalization due to Chomsky, Chomsky (1957):

> If S_1 and S_2 are grammatical sentences, and S_1 differs from S_2 only in that X appears in S_1 where Y appears in S_2 (i.e., $S_1 = \ldots X \ldots$ and $S_2 = \ldots Y \ldots$), and X and Y are constituents of the same type in S_1 and S_2, respectively, then S_3 is a sentence, where S_3 is the result of replacing X by $X + and + Y$ in S_1 (i.e., $S_3 = \ldots X + and + Y \ldots$).

---

*Research Group on Mathematical Linguistics and Language Engineering (GRLMC)

†Institute of Mathematics of the Romanian Academy

‡Research supported by the Secretaría de Estado de Universidades e Investigación – SAB95-0357.

The usual treatment of coordination phenomenon starts from the idea that two categories can be catenated with a conjunction giving a larger category of the same type. The classical rule of this description obeys the context-free requirements, where X can be any linguistic category or nonterminal: $X \rightarrow X$ and X.

This schema can produce the well-known coordination cases between equal categories, which are common in all languages. Several problems arise if we include the treatment of the following cases of coordination:

1. Coordination of *unlike* categories. The general schema for this kind of sentences is similar to $Z \rightarrow Y$ and X, as appearing in the following examples Sag, Gazdar, Wasow, and Weisler (1985):

a. Pat is stupid and a liar (AP and NP).
b. Pat is a Republican and proud of it (NP and AP).
c. Pat is healthy and of sound mind (AP and PP).
d. That was a rude remark and in very bad taste (NP and PP),

where X and Y are different categories or nonterminals and Z is any category resulting from both X and Y. The problem in this schema is to explain how Z is constructed, because it is neither equal to X nor equal to Y (and therefore the rule is not recursive).

2. Binary coordination between many pairs of nonterminals. The following sentences are typical for a language of the form $\{a^n b^m c^n d^m \mid n, m \geq 1\}$, possessing crossed dependencies which cannot be produced by means of context-free rules without regulation:

a. John sent a letter and a postcard to Mary and to Paul, respectively.
b. The boys and the girls eat apples and bananas, respectively.
c. The boys and the girls run and walk through the garden, respectively.
d. *John and Mary sings and dances, respectively.

3. Non-constituent coordination and gapping phenomena. English and other languages contain a number of coordinate constructions where the conjuncts are not constituents in the normal sense but are sequences of constituents. The general term for such constructions is *non-constituent coordination*:

a. Mary studies art, John, music and Paul, history.
b. Harry has sent a letter to Mary, and John, a postcard to Paul.
c. Paul composed, and John posted, a letter to Mary.

Conjunctions can be performed in ordered pairs, where the order of the elements is fixed. The members of a couple may be the same (*o* ... *o* (Spanish), *et* ... *et* (French)) or different (*both* ... *and* (English)). One calls *binary coordination* the structure which has two conjuncts, and *multiple coordination* the structure with more than two. This may express a restriction over the set of conjunctions: *but* cannot appear in multiple coordination, and neither the couples with different words, as in the case of *both* ... *and*. All languages use this kind of structures, and this assumption suggests a unified treatment of the phenomenon. We try to do this in terms of formal operations on strings and languages.

2 Formal Language Prerequisites

As usual, V^* denotes the free monoid generated by the alphabet V with respect to the operation of concatenation; λ stands for the empty string, $V^+ = V^* - \{\lambda\}$, and $|x|$ is the length of $x \in V^*$. The set of all prefixes of $x \in V^*$ is denoted by $Pref(x)$. The *right derivative* of a language $L \subseteq V^*$ with respect to a string $x \in V^*$ is defined by: $\partial^r_x(L) = \{y \in V^* \mid yx \in L\}$. The *shuffle* of two strings $x, y \in V^*$ is defined by: $x \sqcup\!\sqcup\, y = \{u_1v_1 \ldots u_nv_n \mid n \geq 1, x = u_1 \ldots u_n, y = v_1 \ldots v_n, u_i, v_i \in V^*, 1 \leq i \leq n\}$. For $L_1, L_2 \subseteq V^*$ we write $L_1 \sqcup\!\sqcup\, L_2 = \bigcup_{x \in L_1, y \in L_2}(x \sqcup\!\sqcup\, y)$.

A context-free grammar is denoted by $G = (N, T, S, P)$, where N is the nonterminal alphabet, T is the terminal alphabet, $S \in N$ is the axiom, and P is the set of productions, written as $A \rightarrow x$, $A \in N$, $x \in (N \cup T)^*$. The derivation relation is denoted by $\Longrightarrow$, its reflexive and transitive closure by $\Longrightarrow^*$; the language generated by G is denoted by $L(G)$.

The families of regular, linear, context-free, context-sensitive, and recursively enumerable languages are denoted by REG, LIN, CF, CS, and RE, respectively.

A *gsm* ("generalized sequential machine") is a system $g = (K, V_1, V_2, s_0, F, \delta)$, where K is the set of *states*, V_1 is the *input* alphabet, V_2 is the *output* alphabet, $s_0 \in K$ is the *initial state*, $F \subseteq K$ is the set of *final states*, and δ is a mapping (called *transition mapping*) from $K \times V_1$ to the set of finite subsets of $2^{V_2^* \times K}$. We extend δ to $K \times V_1^*$ as follows:

$$\begin{aligned}
&\delta(s, \lambda) = (\lambda, s),\\
&\delta(s, ax) = \{(yx', s') \mid (x', s') \in \delta(s'', x), (y, s'') \in \delta(s, a)\},
\end{aligned}$$

for all $s \in K, a \in V_1, x \in V_1^*$. Then, for $w \in V_1^*, L \subseteq V_1^*$, we define

$$\begin{aligned}
&g(w) = \{z \in V_2^* \mid (z, s_f) \in \delta(s_0, w), \text{ for some } s_f \in F\},\\
&g(L) = \bigcup_{w \in L} g(w).
\end{aligned}$$

A gsm is said to be λ-free if $\delta(s, a) \subseteq V_2^+ \times K$, for all $a \in V_1$, $s \in K$.

It is known that the families in the Chomsky hierarchy are closed under λ-free gsm mappings.

A morphism $h : V^* \rightarrow V^*$ is said to have *limited erasing on a language* $L \subseteq V^*$ (in short, we say that h is *limited on* L) if there is a constant k such that $|x| \leq k|h(x)|$ for all $x \in L, x \neq \lambda$.

All families in the Chomsky hierarchy are closed under limited morphisms.

For further notions and results in formal language theory we use here, we refer to Rozenberg and Salomaa (1997). We only recall here the important notion of a *derivation tree*.

Given a context-free grammar $G = (N, T, S, P)$, a tree τ with the nodes labelled by elements of $N \cup T \cup \{\lambda\}$, is a derivation tree with respect to G if:

1. the root of τ is labelled by S,
2. if the descendents of a node labelled by some $A \in N$ are $\alpha_1, \alpha_2, \ldots, \alpha_k, k \geq 1, \alpha_i \in N \cup T$, then the production $A \rightarrow \alpha_1\alpha_2 \ldots \alpha_k$ is in P,

3. if a node is labelled by λ, then it is the only descendent of a node labelled by some $A \in N$ and $A \rightarrow \lambda$ is a production in P,
4. the nodes labelled by elements of $T \cup \{\lambda\}$ have no descendents, all nodes labelled by elements of N have at least one descendent.

The nodes labelled by elements of $T \cup \{\lambda\}$ constitute the *frontier* of τ (the other nodes – excepting the root – are said to be internal nodes); we denote by $fr(\tau)$ the string in T^* identified by the frontier (we assume τ placed with the root above and we read $fr(\tau)$ from left to right, on the frontier nodes).

For a node ν in τ, we denote by $\tau(\nu)$ the subtree of τ with the root in ν, by $e(\nu)$ the label of ν, and by $fr(\tau(\nu))$ the subword of $fr(\tau)$ corresponding to the frontier of $\tau(\nu)$.

It is known that for every string w generated by a context-free grammar G there is a derivation tree τ with respect to G such that $fr(\tau) = w$; if G is unambiguous, then τ is unique.

3 Two Basic Coordination Operations on Strings

Intuitively, for two strings $x, y \in V^*$ with a common prefix, $x = ux'$, $y = uy'$, the coordination of x, y leads to a string $z = ux'y'$ (or $z = uy'x'$, if the order is not relevant). We consider here two variants of this operation, depending on whether the common prefix is maximal or not.

To start with, let us denote by $mp(x, y)$ the longest common prefix of x, y:

$$\begin{aligned} mp(x, y) = u \quad \text{iff} \quad & x = ux', y = uy' \text{ and there is no } u' \in V^* \\ & \text{such that } x = u'x'', y = u'y'' \text{ and } |u'| > |u|. \end{aligned}$$

Then, the *free prefix coordination* of x, y is defined by:

$$C_{fp}(x, y) = \{ux'y' \mid x = ux', y = uy', \text{ for some } u, x', y' \in V^*\},$$

whereas the *maximal prefix coordination* of x, y is defined by:

$$C_{mp}(x, y) = \{ux'y'\} \text{ iff } x = ux', y = uy' \; u = mp(x, y), x', y' \in V^*.$$

Observe that $C_{fp}(x, y)$ can contain several strings, that always $C_{mp}(x, y) \subseteq C_{fp}(x, y)$ and that $C_{mp}(x, y) \neq \emptyset$ (at least $\lambda \in Pref(x) \cap Pref(y)$; if $mp(x, y) = \lambda$, then $C_{mp}(x, y) = xy$).

Each operation $C_\alpha, \alpha \in \{fp, mp\}$, is extended in the natural way to languages:

$$C_\alpha(L) = \bigcup_{x, y \in L} C_\alpha(x, y).$$

In order to settle the closure properties of languages in the Chomsky hierarchy with respect to operations C_α, we use the following two auxiliary results, relating C_α to known operations on languages.

Lemma 1. *If FL is a family of languages closed under union, concatenation with symbols, intersection with regular languages, and right derivative, then FL closed under $C_\alpha, \alpha \in \{fp, mp\}$, implies FL closed under intersection.*

Proof. For a family FL as above, consider two languages $L_1, L_2 \in FL$, $L_1, L_2 \subseteq V^*$, and construct:

$$L = L_1\{cc_1\} \cup L_2\{cc_2\},$$

where c, c_1, c_2 are symbols not in V. For the regular language $R = V^*\{cc_1c_2\}$ we obtain:

$$L_1 \cap L_2 = \partial_{cc_1c_2}(C_\alpha(L) \cap R), \alpha \in \{fp, mp\}.$$

($\subseteq$) If $x \in L_1 \cap L_2$, then $x_1 = xcc_1 \in L_1, x_2 = xcc_2 \in L_2$. Clearly, $mp(x_1, x_2) = xcc_1c_2 \in C_\alpha(L) \cap R$. Moreover, $x = \partial^r_{cc_1c_2}(xcc_1c_2)$.

($\supseteq$) Take $x \in \partial^r_{cc_1c_2}(C_\alpha(L) \cap R)$. Then $xcc_1c_2 \in C_\alpha(L) \cap R$, hence there are $x_1 \in L_1\{cc_1\}, x_2 \in L_2\{cc_2\}$ such that $xcc_1c_2 \in C_\alpha(x_1, x_2)$. Because xcc_1c_2 contains one occurrence of c_1 and one of c_2, we must have $x_1 = x'_1cc_1, x'_1 \in L_1$, and $x_2 = x'_2cc_2, x'_2 \in L_2$. Because xcc_1c_2 contains only one occurrence of c, it follows that $mp(x_1, x_2) = x_3c$. Consequently, $x'_1 = x'_2 = x_3 = x$. This implies that $x \in L_1 \cap L_2$.

From the closure properties of FL, we obtain that $L \in FL$; if $C_\alpha(L) \in FL$, then also $L_1 \cap L_2 \in FL$. ◊

Lemma 2. *If FL is a family of languages closed under shuffle, λ-free gsm mappings, and limited morphisms, then FL is closed under $C_\alpha, \alpha \in \{fp, mp\}$.*

Proof. Let FL be a family as above and take $L \in FL, L \subseteq V^*$. Consider a new symbol, $c \notin V$. For each $a \in V$, take a new symbol, a', and denote $V' = \{a' \mid a \in V\}$. Consider the morphisms

$h : V^* \longrightarrow V'^*$, defined by $h(a) = a'$, for all $a \in V$,
$h' : (V \cup \{c\})^* \longrightarrow V^*$, defined by $h'(a) = a$, for $a \in V$, and $h'(c) = \lambda$.

Consider also the regular language

$$R = \{aa' \mid a \in V\}^*\{c^2\}V^*V'^*$$

and the gsm

$$g = (\{s_0, s_1, s_2\}, V \cup V' \cup \{c\}, V \cup \{c\}, s_0, \{s_2\}, \delta),$$

with the mapping δ defined by

$$\begin{array}{ll} \delta(s_0, a) = \{(a, s_0)\},\ a \in V, & \delta(s_1, c) = \{(c, s_2)\}, \\ \delta(s_0, a') = \{(c, s_0)\},\ a \in V, & \delta(s_2, a) = \{(a, s_2)\},\ a \in V, \\ \delta(s_0, c) = \{(c, s_1)\}, & \delta(s_2, a') = \{(a, s_2)\},\ a \in V. \end{array}$$

Then we obtain:

$$C_{fp}(L) = h'(g(((L \sqcup\!\sqcup \{c\}) \sqcup\!\sqcup (h(L) \sqcup\!\sqcup \{c\})) \cap R)). \qquad (*)$$

($\subseteq$) If $x \in C_{fp}(L)$, then $x = ux'y'$, for some $x_1 = ux', y_1 = uy'$, both in L. Then $ucx' \in L \sqcup\!\sqcup \{c\}, h(u)ch(y') \in h(L) \sqcup\!\sqcup \{c\}$. If $u = a_1a_2 \ldots a_k, k \geq 0, a_i \in V, 1 \leq i \leq k$, then $a_1a_1'a_2a_2' \ldots a_ka_k'ccx'h(y') \in ((L \sqcup\!\sqcup \{c\}) \sqcup\!\sqcup (h(L) \sqcup\!\sqcup \{c\})) \cap R$.

The gsm g works as follows:

- scanning a prefix $b_1d_1'b_2d_2' \ldots b_rd_r' \in (VV')^*$, one produces $b_1cb_2c \ldots b_rc$,
- when reading cc, these symbols are left unchanged,
- from now one, each $a \in V$ remains unchanged, and each $a' \in V'$ is replaced by a.

Therefore, $g(a_1a_1' \ldots a_ka_k'ccx'h(y')) = a_1c \ldots a_kcccx'y'$. Then, the morphism h' removes all occurrences of c, hence we get the string $a_1 \ldots a_kx'y' = x$.

($\supseteq$) Take a string x in the set in the right-hand side of the $(*)$. There are $x_1 \in L, x_2 \in L, x_3 \in R$ such that x is obtained as in $(*)$ from x_1, x_2, x_3. Denote:

$$\begin{aligned} x_1' &= u_1cx_1'' \in L \sqcup\!\sqcup \{c\}, \text{ for } x_1 = u_1x_1'', \\ x_2' &= h(u_2)ch(x_2'') \in h(L) \sqcup\!\sqcup \{c\}, \text{ for } x_2 = u_2x_2'', \\ x_3 &= a_1a_1' \ldots a_ka_k'ccx_3'x_3'', \text{ for } k \geq 0, x_3' \in V^*, x_3'' \in V'^*. \end{aligned}$$

We must have $a_1a_1' \ldots a_ka_k' \in u_1 \sqcup\!\sqcup h(u_2)$, hence $u_1 = u_2 = a_1 \ldots a_k$. Moreover, $x_1'' = x_3'$ and $h(x_2'') = x_3''$. Consequently, $x_1 = ux_1'', x_2 = ux_2''$, for $u = u_1 = u_2$, and $x = ux_1''x_2''$ (by the definition of g and h'). But $ux_1''x_2'' \in C_{fp}(x_1, x_2) \subseteq C_{fp}(L)$.

According to the closure properties of FL, we get $C_{fp}(L) \in FL$ (note that g is λ-free, h' is limited, and that the closure under gsm mappings – even λ-free – implies the closure under the intersection with regular languages).

For the case of C_{mp} we replace the regular language R by:

$$R' = \{aa' \mid a \in V\}^*\{c^2\}(\{bud'v \mid b, d \in V, u \in V^*, v \in V'^*, b \neq d\} \cup V^* \cup V'^*).$$

As above, we obtain:

$$C_{mp}(L) = h'(g(((L \sqcup\!\sqcup \{c\}) \sqcup\!\sqcup (h(L) \sqcup\!\sqcup \{c\})) \cap R').$$

The intersection with R' forces the selection of strings $a_1a_1' \ldots a_ka_k'ccx'h(y')$ as above, for $a_1 \ldots a_kx' \in L, a_1 \ldots a_ky' \in L$, with maximal k: either the next symbol (the first one in x' and in y') is different in the two strings, or one of x', y' is empty.

Therefore, $C_{mp}(L) \in FL$, too. ◇

Theorem 1. *The families REG, CS, RE are closed under C_α, LIN and CF are not closed, $\alpha \in \{fp, mp\}$.*

Proof. The families in the Chomksy hierarchy have the closure properties in Lemmas 1, 2, but LIN, CF are not closed under intersection. ◇

4 Some Variants of the Basic Operations

In the definition above, either any common prefix or only the maximal prefix of two strings is considered when coordinating the strings. This might not cover the case when only *certain* prefixes can be accepted. This is modeled by the *regulated prefix coordination* operation, defined as follows.

For a regular language $M \subseteq V^*$ and $x, y \in V^*$, we define:

$$C_{rp}(x,y) = \{ux'y' \mid x = ux', y = uy', \text{ for some } u, x', y' \in V^*, u \in M\}.$$

(Only prefixes belonging to M are taken into consideration.)

Another natural variant is to coordinate substrings of the two strings, identifying both prefixes and suffixes of them. Formally, the *free bilateral coordination* of $x, y \in V^*$ is defined by:

$$C_{fb}(x,y) = \{ux'y'v \mid x = ux'v, y = uy'v, \text{ for some } u, v, x', y' \in V^*\}.$$

Because it is not clear how the *maximal bilateral coordination* should be defined, we do not consider here this case. (For instance, consider $x = abbab$, $y = abbbab$. The maximal common prefix is abb, the maximal common suffix is $bbab$; they overlap, both in x and in y!)

We can, however, define in the usual way the *regulated bilateral coordination*, asking that both u, v in the definition above are elements of a given regular language; we denote by C_{rb} this operation.

Both C_{fb}, C_{rb} can be extended in the natural way to languages.

The next step is to iterate the operations $C_\alpha, \alpha \in \{fp, mp, rp, fb, rb\}$, defining, for $L \subseteq V^*$:

$$C_\alpha^*(L) = \bigcup_{i \geq 0} C_\alpha^i(L),$$

where

$$C_\alpha^0(L) = L,\ C_\alpha^{i+1}(L) = C_\alpha(C_\alpha^i(L)),\ i \geq 0.$$

It is easy to see that Lemma 1 holds true with the same proof for all operations C_α, C_α^*, α as above: with the notation in the proof of Lemma 1, we have:

$$C_\alpha^*(L) \cap R = C_\alpha(L) \cap R, \alpha \in \{fp, mp, fb\},$$

because the intersection with R selects the strings where only one occurrence of c is present. Moreover, for

$$R = M\{cc_1c_2\},$$

we also cover the case of $\alpha \in \{rp, rb\}$.

Theorem 2. *The families LIN, CF are closed under none of the operations C_α, C_α^*, $\alpha \in \{fp, mp, rp, fb, rb\}$.*

Let us examine now the proof of Lemma 2.

Instead of transforming a prefix $a_1a_1' \ldots a_ka_k'$ of the scanned string into $a_1a_2 \ldots a_k$, the gsm g can also check whether or not $a_1a_2 \ldots a_k \in M$, for a given regular set. (Take a finite automaton for M and simulate it on the symbols $a_1, \ldots, a_k$; the details are left to the reader.) Therefore, Lemma 2 holds true also for C_{rp}.

For the bilateral case, we modify the proof of Lemma 2 as follows:

- consider three new symbols c_1, c_2, c,
- instead of R, consider the regular set:

$$R' = \{aa' \mid a \in V\}^*\{c_1^2\}V^*V'^*\{c_2^2\}\{aa' \mid a \in V\}^*.$$

– instead of g, consider the gsm:

$$g' = (\{s_0, s_1, s_2, s_3, s_4\}, V \cup \{c_1, c_2\}, V \cup \{c\}, s_0, \{s_4\}, \delta'),$$

with the transition mapping defined as suggested by Figure 1.

Then, for $L \subseteq V^*$ we obtain:

$$C_{fb}(L) = h'(g'(((L \sqcup\!\sqcup \{c_1c_2\}) \sqcup\!\sqcup (h(L) \sqcup\!\sqcup \{c_1c_2\})) \cap R')).$$

If the coordination is regulated by a language $M \in REG$, then, as in the case of C_{rp}, we can modify g' above in such a way to check whether or not the used prefix and suffix of the current strings are in M.

Consequently, Lemma 2 holds true also for C_α, $\alpha \in \{fb, rb\}$.

Theorem 3. *The families REG, CS, RE are closed under all operations C_α, $\alpha \in \{fp, mp, rp, fb, rb\}$.*

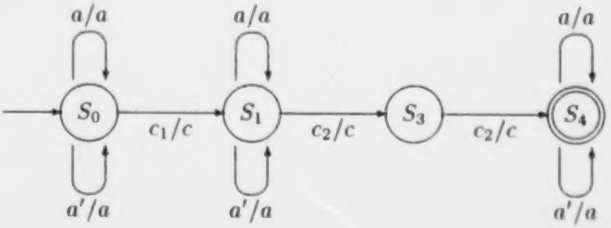


Figure 1

Theorem 4. *The families CS, RE are closed under all operations C_α^*, $\alpha \in \{fp, mp, rp, fb, rb\}$.*

Proof. For RE, the assertion is obvious (consequence of the Turing-Church Thesis). For CS, a straightforward (but long) construction can prove the assertion.

Here is the idea of such a construction for C_{fp}^*. The modifications for the other cases are obvious:

- Start from a context-sensitive grammar G, for a language $L \subseteq V^*$;
- Generate a string $x \in L(G)$;
- Generate one more string $y \in L(G)$;
- Find a common prefix of x, y; let it be u (hence $x = ux', y = uy'$);
- Having x, y, construct $ux'y'$;
- Consider $ux'y'$ in the role of x and go to step 2.

It is clear that the grammar G' obtained in this way generates exactly $C_{fp}^*(L)$. Moreover, G' has a bounded workspace: in order to generate a string w, it uses at most a space of length $2|w|$ (from $x = ux', y = uy'$ we get $w = ux'y'$ and $|xy| \leq 2|ux'y'|$). Consequently, $L(G') \in CS$. ◇

The closure of REG under the operations $C_\alpha^*, \alpha \in \{fp, mp, fb, rb\}$, remains open.

5 Syntactically Grounded Coordination

In the previous sections we have defined coordination operations on strings x, y without taking into account the syntactic structure of x, y. In natural languages, we pass from $x = ux', y = uy'$ (or $x = ux'v, y = uy'v$) to $w = ux'y'$ ($w = ux'y'v$, respectively) only when u (u, v, respectively) has (have) the same syntactic structure. This makes necessary to consider the derivation trees of x, y, hence to define the coordination for trees, not for strings.

Take a context-free grammar $G = (N, T, S, P)$.

Two derivation trees τ_1, τ_2 with respect to G are said to be *coordinable* if there is a node ν_1 in τ_1 and a node ν_2 in τ_2 such that if we remove $\tau_1(\nu_1)$ from τ_1 and $\tau_2(\nu_2)$ from τ_2 and we label both ν_1 and ν_2 with the same symbol, then we obtain two identical trees.

For two coordinable trees τ_1, τ_2 with respect to nodes ν_1, ν_2 labelled by X, Y, respectively, we construct a tree τ_3 as follows:

1. Excise $\tau_1(\nu_1)$ from τ_1.
2. Label ν_1 in the remaining tree by a new nonterminal symbol, Z.
3. Add at this node the subtree defined by the context-free rule $Z \rightarrow XY$.
4. Attach $\tau_1(\nu_1)$ to the new node labelled by X and $\tau_2(\nu_2)$ to the new node labelled by Y.

The obtained tree, τ_3, has a terminal frontier.

If we have $fr(\tau_1) = u_1x_1v_1$, $fr(\tau_2) = u_2x_2v_2$, for $x_1 = fr(\tau_1(\nu_1))$, $x_2 = fr(\tau_2(\nu_2))$, then, because τ_1, τ_2 are coordinable with respect to ν_1, ν_2, we have $u_1 = u_2$, $v_1 = v_2$. Then, $fr(\tau_3) = ux_1x_2v$, for $u = u_1 = u_2, v = v_1 = v_2$.

Therefore, $fr(\tau_3) \in C_{fb}(fr(\tau_1), fr(\tau_2))$.

We say that τ_3 has been obtained by coordination from τ_1, τ_2. For a grammar G, we denote by $C(G)$ the language consisting of all strings $fr(\tau)$, for τ being a tree obtained by coordination from two derivation trees with respect to G.

Note that we define $C(G)$ using exactly one coordination for each string in $C(G)$.

Figure 2 presents the idea of coordinable trees and of coordination.

The fact that, when coordinating (the frontier of) trees, the common parts of the frontiers are not only equal but they also have the same syntactic description has a rather powerful (and somewhat surprising) influence on the result: the operation preserves context-freeness.

Theorem 5. *For every context-free grammar G, the language $C(G)$ is context-free.*

Proof. If $G = (N, T, S, P)$, then we construct the grammar $G' = (N \cup \{Z\}, T \cup \{c\}, S, P')$, with

$$\begin{aligned} P' &= P \cup \{A \rightarrow xZy \mid A \rightarrow xXy \text{ and } A \rightarrow xYy \in P, \\ &\qquad \text{for some } x, y \in (N \cup T)^*, X, Y \in N\} \\ &\cup \{Z \rightarrow XcY \mid X, Y \in N\}. \end{aligned}$$

Consider also the regular set:

$$R = T^*\{c\}T^*$$

and the morphism $h : (T \cup \{c\})^* \to T^*$ defined by $h(a) = a$, $a \in T$, and $h(c) = \lambda$. We obtain:

$$C(G) = h(L(G') \cap R).$$

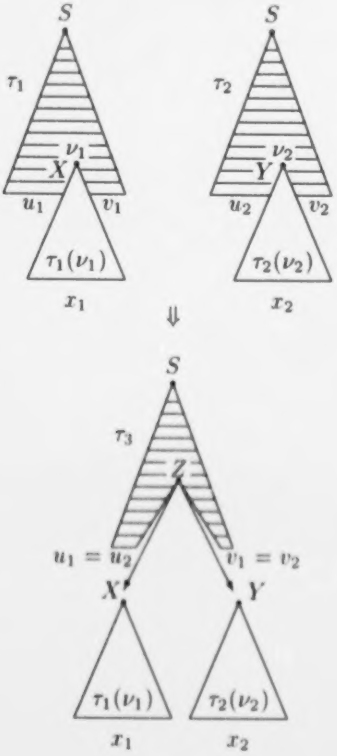


Figure 2

Indeed, because when we coordinate two trees τ_1, τ_2 of G with respect to two nodes ν_1, ν_2 the trees obtained by excising $\tau_1(\nu_1), \tau_2(\nu_2)$ from τ_1, τ_2, respectively, are identical modulo the labelling of ν_1, ν_2, there is a rule $A \to xXy$ used in τ_1 and a rule $A \to xYy$ used in τ_2 (possibly $X = Y$). Therefore, the new rules of P' perform a coordination operation. Removing c by the morphism h, we get the result of coordination, hence $C(G) \subseteq h(L(G') \cap R)$. Because the intersection with R selects from $L(G')$ exactly those strings in whose derivation we have used only once a rule $Z \to XcY$, we have also the converse inclusion.

From the closure properties of CF we obtain $C(G) \in CF$. ◊

Corollary. *If G is a regular grammar, then $C(G)$ is also regular.*

Proof. Exactly as above, starting from G regular, we construct G'. Because all recurrent derivations $A \Longrightarrow^* uAv$ in G' have $u \in T^*, v = \lambda$, according to Theorem 5.5 in Salomaa (1973), it follows that $C(G)$ is a regular language. ◊

For linear grammar, the above results are not true.

Theorem 6. *There is a linear grammar G such that $C(G) \notin LIN$.*

Proof. For the grammar:

$$G = (\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow ab\}),$$

we clearly obtain:

$$C(G) = \{a^i a^n b^n a^m b^m b^i \mid i \geq 0, n, m \geq 1\},$$

which is not a linear language. ◊

In the coordination operation defined above, we allow not only to have both nodes ν_1, ν_2 identically labelled, but it is also possible to have $\tau_1(\nu_1) = \tau_2(\nu_2)$ (not to mention the weaker condition, $fr(\tau_1(\nu_1)) = fr(\tau_2(\nu_2))$). Coordinating τ_1 with τ_1 looks artificial. Hence, we say that the coordination of τ_1, τ_2 with respect to the nodes ν_1, ν_2 is *nontrivial* if $\tau_1(\nu_1) \neq \tau_2(\nu_2)$. We denote by $NC(G)$ the language consisting of all strings $fr(\tau)$, for τ being obtained by a nontrivial coordination of two derivation trees with respect to G.

The apparently small difference between usual tree coordination and nontrivial coordination turns out to have a surprising effect on the type of the language $NC(G)$.

Theorem 7. *There is a linear grammar G such that $NC(G) \notin CF$.*

Proof. Consider again the grammar G in the proof of Theorem 6. All derivations in G are of the form:

$$S \Longrightarrow aSb \Longrightarrow a^2Sb^2 \Longrightarrow \ldots \Longrightarrow a^nSb^n \Longrightarrow a^{n+1}b^{n+1}, \; n \geq 0.$$

Thus, two subtrees of derivation trees with respect to G are different if they have different heights. Therefore, we have:

$$NC(G) = \{a^i a^n b^n a^m b^m b^i \mid i \geq 0, n, m \geq 1, n \neq m\}.$$

Let us assume that $NC(G) \in CF$. Consider a context-free grammar $G' = (N, \{a, b\}, S, P)$ such that $L(G') = NC(G)$. In order to generate the prefix a^i and the suffix b^i in strings $a^i a^n b^n a^m b^m b^i$ of $NC(G)$, we need a derivation $X \Longrightarrow^* a^j X b^j$, $j \geq 1, X \in N$. The rules used in such a derivation are not used when producing substrings $a^n b^n$ or $a^m b^m$, because from X we have to generate substrings of the form $a^k a^n b^n a^m b^m b^l$. If in a subderivation leading to a block $a^n b^n$ or $a^m b^m$ of a string $a^i a^n b^n a^m b^m b^i$, after generating some $a^s Y b^s$, we introduce X from Y, then strings not in $NC(G)$ will be obtained. Therefore, if we remove from P all rules contributing to a derivation $X \Longrightarrow^* a^j X b^j$ as above, then we obtain a grammar

G'' generating strings of the form $a^{i_1}a^nb^na^mb^mb^{i_2}$, for all $n, m \geq 1$, $n \neq m$, and with finitely many values for i_1, i_2. Because the pairs (i_1, i_2) are well specified and finitely many, we can replace by λ each occurrence of a and b in the rules of G'' which contribute to the prefix a^{i_1} and to the suffix b^{i_2}, respectively. In this way, we obtain a context-free grammar G''' such that $L(G''') = \{a^nb^na^mb^m \mid n, m \geq 1, n \neq m\}$.

However, $L = \{a^nb^na^mb^m \mid n, m \geq 1, n \neq m\}$ is not a context-free language. Assume the contrary. It follows that also $L' = \{a^nb^nca^mb^m \mid n, m \geq 1, n \neq m\}$ is context-free. Take a context-free grammar G_0 for L'. All recurrent derivations in G_0 must be of the form $X \Longrightarrow^* a^iXb^i, i \geq 0$. (Any other type of recurrent derivations produces strings not in $a^nb^na^mb^m$, even without imposing restrictions on the relation between n and m.) Replace by λ each occurrence of b in G_0. The obtained grammar, G_0', contain only recurrent derivations of the form $X \Longrightarrow^* a^iX, i \geq 0$. According to Theorem 5.5 in Salomaa (1973), the language $L(G_0')$ must be regular. However, $L(G_0') = h(L')$, for $h(a) = a, h(c) = c, h(b) = \lambda$. Hence, $L'(G_0') = \{a^nca^m \mid n, m \geq 1, n \neq m\}$. This is not a regular language, a contradiction which concludes the proof. ◇

Theorem 8. *There is a regular grammar G such that $NC(G) \notin REG$.*

Proof. Consider the grammar:

$$G = (\{S\}, \{a, b\}, S, \{S \to aS, S \to b\}).$$

We obtain:

$$NC(G) = \{a^ia^nba^mb \mid i \geq 0, n, m \geq 0, n \neq m\}.$$

As in the previous proof, if $NC(G)$ is regular, then a regular grammar generating $\{a^nba^mb \mid n, m \geq 0, n \neq m\}$ can be constructed, which is contradictory, because this language is not regular. It follows that $NC(G) \notin REG$, too. ◇

6 Central Coordination

The case of *The boys eat apples + The girls eat bananas → The boys and the girls eat apples and bananas (respectively)* suggests the following variants of coordination operations.

For $x, y \in V^*$, we define the *free central coordination* by:

$$C_{fc}(x, y) = \{x'y'ux''y'' \mid x = x'ux'', y = y'uy'', \text{ for some } u, x', x'', y', y'' \in V^*\},$$

whereas the *maximal central coordination* is defined by:

$$\begin{aligned} C_{mc}(x, y) = \{x'y'ux''y'' \quad \mid \quad & x = x'ux'', y = y'uy'', \text{ for } u, x', x'', y', y'' \in V^*, \\ & \text{and there is no proper superword of} \\ & u \text{ which is common to } x \text{ and } y\}. \end{aligned}$$

The proof of Lemma 1 holds true also for C_α, C_α^*, for $\alpha \in \{fc, mc\}$, whereas the proof of Lemma 2 can be modified in order to cover the new operation (see also the remarks before Theorem 3). We get:

Theorem 9. *The families REG, CS, RE are closed under C_α. CS, RE are closed under C_α^*, too, but LIN, CF are closed under none of these operations,* $\alpha \in \{fc, mc\}$.

The central coordination can be naturally defined in the syntactically grounded variant, imposing that the substring u has the same syntactical description in both strings x, y.

We denote by $C_c(G)$ the language of the frontier strings of trees obtained by central coordination of derivation trees in the context-free grammar G. Somewhat surprisingly, for this case we do not have a result like Theorem 4 above.

Theorem 10. *There is a linear grammar G such that $C_c(G) \notin CF$.*

Proof. Consider the grammar:

$$G = (\{S, A, B, C\}, \{a, b, c, d, e\}, S, P),$$
$$P = \{S \rightarrow A, S \rightarrow B, A \rightarrow aAb, B \rightarrow cBd, A \rightarrow C, B \rightarrow C, C \rightarrow e\}.$$

All strings of the form $x = a^n e b^n$, $y = c^m e d^m$, $n, m \geq 0$, are in $L(G)$ and they have the (maximal) common subword e. Moreover, in any derivation tree there is the subtree determined by $C \rightarrow e$. Thus we have:

$$C_c(G) \cap \{a, c\}^* e \{b, d\}^* = \{a^n c^m e b^n d^m \mid n, m \geq 0\},$$

a language which is not context-free. ◊

One can easily see that for each regular grammar G we have $C_c(G) \in REG$: two derivation trees with respect to G can have in common only a subtree corresponding to a final part of the associated derivations, so the central coordination is, in fact, a "suffix coordination".

7 Concluding Remarks

The following table synthesizes the results concerning the closure properties of families in the Chomsky hierarchy under the (non-iterated) coordination operations considered above (the notations are those used before, U stands for "undefined")

	C_{fp}	C_{mp}	C_{rp}	C_{fb}	C_{rb}	C	NC	C_{fc}	C_{mc}	C_c
REG	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
LIN	No	No	No	No	No	No	No	No	No	No
CF	No	No	No	No	No	Yes	No	No	No	No
CS	Yes	Yes	Yes	Yes	Yes	U	U	Yes	Yes	U
RE	Yes	Yes	Yes	Yes	Yes	U	U	Yes	Yes	U

Several interesting conclusions can be drawn on the basis of the results in this table.

With only one exception, that of unrestricted syntactically grounded coordination, none of these operations preserves the context-free languages.

Coordination is a basic linguistic operation, probably present in all languages. We have considered here many variants, both at the surface level of strings and taking into consideration derivation trees, hence the *structure* of strings. Thus, we can claim, at least statistically, that we have captured the idea of coordination by our definitions. Consequently, we can infer: *natural languages contain non-context-free specific constructions.* Coordination is one of them.

This does not necessarily implies that natural languages, English for instance, are not context-free languages in the restricted sense, as sets of strings, it merely means that *natural languages contain constructions which cannot be handled by the context-free grammar formalism.*

The only case when context-free languages are preserved, that of unrestricted syntactically grounded coordination, can be considered as insufficiently adequate, as it allows trivial coordination. Significantly enough, when nontrivial coordination is considered, the context-freeness is again lost. Moreover, there are linear grammars leading to non-context-free languages by nontrivial syntactically grounded coordination. (The fact that the family of linear languages is closed under none of the considered operations is not a surprise, because this family is not closed under concatenation while coordination involves a sort of concatenation in its definition.)

If context-free grammars are not sufficient, then what else ? This is not an easy question. The family of context-sensitive languages is closed under all coordination operations (except those syntactically grounded, which make no sense in this case as we have no available derivation tree). This confirms the general belief that natural languages lie somewhere at the context-sensitive level of the Chomsky hierarchy, but this is a loose conclusion: context-sensitive grammars are "too powerful". A standard candidate are the Tree Adjoining Grammars (TAG), of various forms (with constraints, dependencies, etc). How the corresponding families of languages behave with respect to the previous coordination operations (on strings or on trees) remains as a *research topic.* The answer is of a definite interest for the adequacy of TAG's as models of the syntax of natural language.

References

Chomsky, N. (1957). *Syntactic structures.* Mouton, The Hague.

Rozenberg, G. and A. Salomaa (Eds.) (1997). *Handbook of formal languages.* Springer-Verlag.

Sag, I., G. Gazdar, T. Wasow, and S. Weisler (1985). Coordination and how to distinguish categories. *Natural Language and Linguistic Theory 3*, 117–171.

Salomaa, A. (1973). *Formal languages.* Academic Press, New York.

Improving the Precision of a Text Retrieval System with Compound Analysis

Renée Pohlmann*
Wessel Kraaij†

Abstract

In this paper we describe research on compound analysis in the UPLIFT information retrieval project. Results of earlier experiments indicated that splitting up compounds in the query and forming new compounds by combining query terms improves recall while precision does not deteriorate. We investigated whether adding syntactic constraints to the compound splitting and formation processes would improve our initial results. We compared different strategies for compound formation and we also investigated the effect of adding compound constituents as separate index terms. The results of our experiments show that using information about head-modifier relationships to create complex index terms can improve both recall and precision significantly but only if all constituents are also added separately. We found that using both noun-adjective and noun-noun head-modifier pairs produced the best results.

Introduction

The work described in this paper is part of the UPLIFT project[1]. UPLIFT investigates whether linguistic tools can improve and extend the functionality of vector space text retrieval systems (cf. Salton (1989), p. 312 *ff*). Earlier experiments in the UPLIFT project focussed on improving recall by using stemming algorithms[2]. This paper describes an experiment with syntactic phrase indexing techniques for Dutch texts, aimed at improving precision as well as recall. The basic idea behind phrase indexing is that phrases characterize document content more effectively than single word terms. When a single word index is used, a query containing the phrase *information retrieval* will also match with documents containing only *information* or *retrieval*. If *information retrieval* is recognized as a unit, however, these matches may be avoided or given a much lower score (depending on the matching strategy). Different strategies have been used to identify suitable

---

*Utrecht Institute of Linguistics OTS

†Netherlands Organization for Applied Scientific Research (TNO), Institute of Applied Physics

[1] UPLIFT: Utrecht Project: Using Linguistic Information for Free Text retrieval. UPLIFT Home Page: *http://www-uilots.let.ruu.nl/~uplift*

[2] See Kraaij and Pohlmann (1996) for details.

phrases for indexing, the most important distinction being between strategies based on statistical co-occurrence data and strategies based on syntactic processing. So far, both types of strategies have proven to be equally successful (cf. e.g. Fagan (1987), Salton et al. (1990) and, more recently, Hull et al. (1997)). Results of earlier experiments in the UPLIFT project motivated us to take compounds as a starting point for our experimentation with phrase indexing. Our approach was further inspired by the work of Strzalkowski, as described in Strzalkowski (1995) and Strzalkowski and Perez Carballo (1996). Strzalkowski uses syntactic information to identify phrases in queries and documents. These phrases are subsequently normalized (i.e. semantically similar but syntactically different constructions, e.g. *retrieval of information* vs. *information retrieval*, are represented identically) as head-modifier pairs. Other recent work on syntactic phrase indexing includes Evans and Zhai (1996) and Smeaton et al. (1995). In sections 1 to 5 we describe our approach and discuss the set-up and the results of the experiments. In section 6 we present the conclusions and give some possibilities for further research.

1 Compounds and related constructions

Earlier research in the UPLIFT project showed that when a query is expanded with the constituents of compounds already occurring in it[3] and new compounds are added to the query by combining query terms, recall improves while precision does not deteriorate. The following example illustrates this approach.

> Query: *Ik zoek documenten over computers en natuurlijke taalverwerking*
> ("I am looking for documents on computers and natural language processing")

This query would result in the following index terms (after removal of stop words):

document	
computer	
natuurlijk	
taalverwerking	
taal	compound splitting
verwerking	"
computertaal	compound formation
taalcomputer	"

In the example, the compounds *computertaal* (computer language) and *taalcomputer* (language computer) are added to the query by combining *computer* and *taal*. Both are valid compounds[4] but, although the second compound may retrieve relevant articles for this query, the first (a synonym for programming language) will probably retrieve many unrelated documents.

---

[3] In Dutch, compounds are usually written as a single orthographic unit, e.g. *levensverzekeringsmaatschappij* (life insurance company). As a result of this, compound constituents are normally not considered as separate index terms.

[4] New compounds are validated using a list of all the compounds found in the document collection.

We decided to investigate whether it would be possible to improve precision as well by using syntactic information to constrain the compound splitting and compound formation processes.

We restricted compound splitting by creating system variants which only add the heads or both heads and modifiers as separate index terms. To split up compounds into their constituents we used the dictionary-based compound splitter developed by Theo Vosse for the CORRie project (cf. Vosse (1994)). The compound splitter does not assign structure to the compound but simply yields a list of constituents. Identifying head-modifier relationships in compounds is not trivial because of possible structural ambiguities. In Dutch, compounds existing of two parts are usually right-headed (a *fietswiel* is a type of *wiel*) but compound construction is recursive and both the head and the modifier can be compounds themselves, resulting in structural ambiguities, e.g. [[X1 X2] X3] or [X1 [X2 X3]]. We have not attempted to implement a strategy to solve all structural ambiguities in compounds but we have applied two different heuristics to assign probable structures. In a recent study, ter Stal (1996) found that simply assuming that all compounds have a left-branching structure produced ± 70% correct results. Although his results are for English, we decided to try this strategy. As an alternative, we also implemented a strategy where we use unambiguous cases collected from the corpus to confirm a certain choice. If we find independent evidence for a left-branching structure (X1 modifies X2 in unambiguous contexts) or a right-branching structure (X1 modifies X3) we select the appropriate structure. If we do not find independent evidence for either structure we choose a left-branching structure by default[5].

The formation of new compounds was restricted by using only terms which occur in a certain syntactic context to generate new complex terms. We restricted compound generation to term pairs originating from complex Noun Phrases (NPs) containing a specific type of Prepositional Phrase (PP) (with the preposition *van*, *voor* or *door*) as a noun post-modifier. The choice for this construction was motivated by the fact that many compounds in Dutch can be paraphrased using a specific type of PP, e.g. *fietswiel* (bicycle wheel) ↔ *wiel van een fiets* (wheel of a bicycle), see, for instance, Geerts et al. (1984) p. 103. The term pairs were created by combining the head noun of the main NP with the head noun of the NP contained in the PP. Figure 1 illustrates this process. PP-modification structures exhibit similar ambiguities to the ones in complex compounds, e.g. in *the man with the dog with the spots* it is not clear whether the PP *with the spots* modifies *the man* or *the dog*. We decided to treat these structures analogously to the compound structures. The default strategy we adopted was to assume that PP modification structures are right-branching (i.e. each PP modifies the noun immediately preceding it). We again also implemented a second strategy, using corpus data for disambiguation. We later extended compound generation by using all PP post-modifiers and adjective pre-modifiers as well.

To ensure matching, both original and new complex terms were normalized as head-modifier pairs. Complex constructions consisting of more than 2 constituents are represented as several head-modifier pairs. See figure 2 for an example.

---

[5] A third option, where the structure is simply left ambiguous and all interpretations are selected, was not implemented for lack of time.

Furthermore, both queries and documents were treated analogously. This was not possible in our earlier approach (combining all the terms in a document to create compounds is clearly not feasible). In this way we ensured that matches between compounds and equivalent constructions would be given the same score as literal matches.

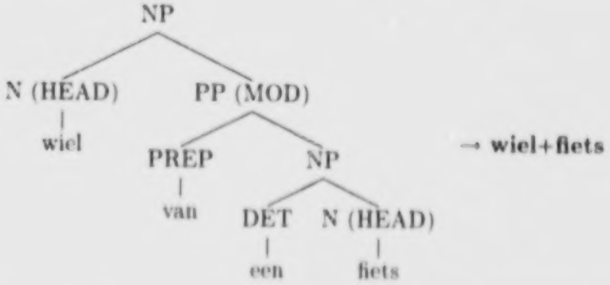


Figure 1: Term pair extraction from NP with PP modifier

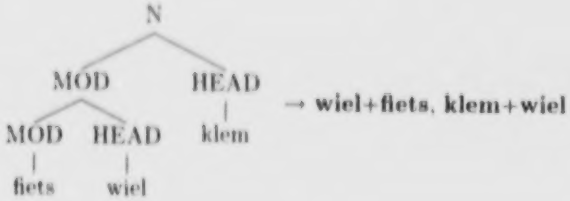


Figure 2: Term pair extraction from complex compound

2 Indexing module

Based on the options described in section 1 above we developed several different versions of an indexing module and integrated each of these with our retrieval engine[6] to create different system variants. The indexing modules consist of the following basic sub-routines.

A string segmentation algorithm (**tokenizer**) is used to identify sentence and word boundaries.

A **lexical look-up** algorithm, based on the CELEX lexical database for Dutch (Baayen et al. (1993)) assigns part-of-speech tags to the words.

A **tagger** is used to resolve ambiguities in tag assignment. We used the Multext tagger, (cf. Armstrong et al. (1995)), a Hidden Markov Model tagger, which has

---

[6] The retrieval engine used in the UPLIFT project is the TRU vector space engine developed by Philips Research (cf. Aalbersberg et al. (1991)).

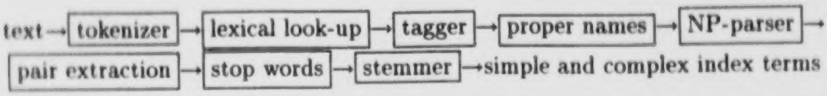


Figure 3: Indexing process

the advantage that it requires only a partially disambiguated corpus for training. After training, the tagger produced 91.5% correct results.

A very simple heuristic based on the distinction upper case–lower case is used to glue sequences of **proper names** together (e.g. *Verenigde_Staten_van_Amerika* (United States of America)). In this way we ensure that proper names are treated as a unit and term pairs are not extracted from them.

An **NP-parser** is used to identify NPs in the texts. The parser we use was developed by TNO-TPD, (cf. van Surksum and den Besten (1993)). This parser is deterministic and requires fully disambiguated input from the tagger. It is also robust and fast (244 sentences per second on a Sun-Sparc 10/40). The coverage of the NP-grammar is not complete[7], but for the purpose of our experiment it was considered to be sufficient.

Since the parser is deterministic and only generates one analysis for ambiguous structures, separate **pair extraction** modules extract the appropriate word pairs and single words from the output of the parser.

A **stop word list** is used to identify and eliminate so-called stop words (mostly function words).

Finally, all remaining words (and compound constituents) are replaced by their stem using a dictionary-based (CELEX) **stemming** algorithm. We used the best variant of all the stemming algorithms tested in previous UPLIFT experiments (cf. Kraaij and Pohlmann (1996)). This variant handles inflection only. Figure 3 shows how the different sub-routines work together.

3 System variants

We developed and tested a large number of system variants (23). These variants are summarized below. The names are abbreviations of the type vABC which must be interpreted as follows: A refers to the syntactic context from which of the head-modifier pairs are generated, B to the strategy used for the disambiguation of complex structures and C to the treatment of constituents of complex structures.

vXXX	No compound analysis but tagging, proper name identification and stemming are included. We added this version to see whether tagging, stemming and proper name recognition alone would already be sufficient to improve precision.
vM..	Head-modifier pairs are generated from compounds.

---

[7] Relative clauses, for instance, are not included.

vS.. Head-modifier pairs from complex NPs with specific PP post-modifiers (see section 1 above) are added.

vP.. Head-modifier pairs from all PP post-modifiers are added.

vA.. Head-modifier pairs from adjective pre-modifiers are added.

vAP.. A combination of the two previous versions.

v.a. Complex terms are analyzed using the default strategies.

v.c. Complex terms are analyzed using corpus data.

v..1 All constituents of complex terms are also added separately to the index.

v..2 Only heads (including heads of complex modifiers) are added to the index separately.

v..3 Only the head of the entire complex construction is added as a separate index term.

v..4 Constituents are not added separately.

We compared these variants with the following two versions:

vn Baseline. TRU retrieval engine, no extensions.

c4fow Best version from previous experiments, all constituents of compounds are added to the query and new compounds are generated by arbitrarily combining query terms.

4 Test procedures

The test collection used for the experiments was compiled during previous research in the UPLIFT project on stemming algorithms. It consists of a document collection of 59,608 articles published in *Het Eindhovens Dagblad*, *Het Brabants Dagblad* and *Het Nieuwsblad* from January to October 1994 and 36 queries and relevance judgements. Some general statistics for the document collection are given in table 1 below.

Total number of documents	59,608
Total number of words (tokens)	26,585,168
Total number of terms (types)	434,552
Max number of words per document	5,979
Av. number of words per document	446
Max number of terms per document	2,291
Av. number of terms per document	176

Table 1: Document collection statistics

The queries were formulated by test subjects recruited among staff and students of Utrecht University. Test subjects also performed the relevance judgements for their queries.

Retrieval performance is usually evaluated using measures derived from the following two main parameters:

$$Recall = \frac{\text{number of relevant items retrieved}}{\text{total number of relevant items in collection}}$$

$$Precision = \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}}$$

The values for recall and precision range from 0 (low) to 1 (high). When precision is high, recall is usually low and vice versa.

The computation of recall is a traditional problem in IR evaluation. It is impossible to estimate the total number of relevant items in a document collection for a certain query without doing relevance assessments for nearly the complete collection. The common solution to this problem is to use the so-called *pooling method*[8]. This method is based on the assumption that if one uses a variety of different retrieval systems to create a document *pool* for each query, the probability that most relevant documents will be contained in the pool is high. The list of relevant documents for each query is then compiled by judging only those documents contained in the pool.

We used 4 different derived measures to evaluate retrieval performance for this experiment. These measures are: average precision, ap5-15 (precision at 5, 10 and 15 documents retrieved, averaged), R-recall (recall at R, where R is the number of relevant articles for a particular query) and recall1000 (recall at 1000 documents retrieved). Average precision and R-recall measure general performance for precision and recall respectively. The ap5-15 measure should give an idea of the performance of the system variants for shallow searches where only the first few documents will be considered and the recall1000 measure is aimed at more in-depth searches. We also performed statistical significance tests to establish whether the differences between values are significant or should be attributed to chance. The design chosen for these statistical tests is based on Tague-Sutcliffe (1995a) and Tague-Sutcliffe (1995b). Details on the statistical tests can be found in the appendix.

5 Results

The results of the experiment are summarized in table 2. The percentages indicate improvement/decrease compared to the performance of the baseline (vn)[9]. The results of the statistical significance tests are summarized in tables 3, 4, 5 and 6. In these tables system versions have been divided into equivalence classes indicated by numbers.

The results show that tagging, proper name recognition and stemming alone are not sufficient to improve average precision significantly (vXXX is assigned to the

---

[8] See Harman (1993), p. 9 *ff*.

[9] Note that figures have been rounded. This accounts for small differences between seemingly equivalent versions.

version	avp	% change	ap5-15	% change
vXXX	0.330 (0.218)	+ 5.4	0.420 (0.285)	+ 8.4
vMa1	0.350 (0.215)	+ 11.9	0.443 (0.284)	+ 14.3
vMc1	0.350 (0.215)	+ 11.8	0.444 (0.284)	+ 14.4
vMa2	0.340 (0.225)	+ 8.7	0.448 (0.309)	+ 15.6
vMc2	0.341 (0.225)	+ 9.0	0.446 (0.308)	+ 15.0
vMa3	0.344 (0.227)	+ 9.9	0.451 (0.311)	+ 16.3
vMc3	0.344 (0.227)	+ 9.9	0.450 (0.311)	+ 16.2
vMa4	0.330 (0.212)	+ 5.4	0.427 (0.274)	+ 10.0
vMc4	0.330 (0.212)	+ 5.6	0.426 (0.274)	+ 9.9
vSa1	0.346 (0.214)	+ 10.7	0.441 (0.283)	+ 13.9
vSc1	0.348 (0.213)	+ 11.3	0.443 (0.282)	+ 14.3
vSa2	0.284 (0.240)	− 9.3	0.366 (0.334)	− 5.7
vSc2	0.286 (0.239)	− 8.6	0.365 (0.331)	− 5.8
vSa3	0.285 (0.240)	− 8.9	0.361 (0.335)	− 6.9
vSc3	0.286 (0.238)	− 8.6	0.363 (0.331)	− 6.3
vSa4	0.277 (0.236)	− 11.3	0.349 (0.329)	− 10.0
vSc4	0.279 (0.235)	− 10.8	0.349 (0.329)	− 10.0
vPa1	0.354 (0.221)	+ 13.2	0.451 (0.282)	+ 16.2
vAa1	0.347 (0.210)	+ 11.1	0.443 (0.277)	+ 14.3
vAa4	0.309 (0.216)	− 1.0	0.383 (0.277)	− 1.3
vAPa1	0.358 (0.217)	+ 14.6	0.448 (0.276)	+ 15.7
c4fow	0.319 (0.200)	+ 2.2	0.427 (0.277)	+ 10.0
vn	0.313 (0.214)		0.388 (0.291)	
version	**R-recall**	**% change**	**recall1000**	**% change**
vXXX	0.316 (0.206)	+ 12.1	0.855 (0.166)	+ 11.8
vMa1	0.344 (0.186)	+ 22.1	0.914 (0.102)	+ 19.5
vMc1	0.343 (0.186)	+ 22.0	0.914 (0.102)	+ 19.5
vMa2	0.313 (0.206)	+ 11.2	0.879 (0.136)	+ 14.8
vMc2	0.312 (0.207)	+ 10.8	0.873 (0.145)	+ 14.1
vMa3	0.312 (0.207)	+ 11.0	0.875 (0.141)	+ 14.4
vMc3	0.312 (0.207)	+ 11.0	0.875 (0.141)	+ 14.3
vMa4	0.311 (0.201)	+ 10.3	0.850 (0.165)	+ 11.1
vMc4	0.310 (0.201)	+ 10.3	0.849 (0.167)	+ 10.9
vSa1	0.343 (0.184)	+ 21.7	0.918 (0.104)	+ 19.9
vSc1	0.343 (0.184)	+ 21.8	0.918 (0.104)	+ 19.9
vSa2	0.260 (0.212)	− 7.7	0.783 (0.230)	+ 2.4
vSc2	0.260 (0.210)	− 7.6	0.778 (0.232)	+ 1.7
vSa3	0.254 (0.216)	− 9.6	0.758 (0.246)	− 0.9
vSc3	0.256 (0.214)	− 9.2	0.760 (0.241)	− 0.7
vSa4	0.246 (0.212)	− 12.5	0.746 (0.249)	− 2.5
vSc4	0.248 (0.210)	− 11.8	0.746 (0.247)	− 2.6
vPa1	0.348 (0.189)	+ 23.6	0.919 (0.100)	+ 20.2
vAa1	0.343 (0.188)	+ 21.8	0.920 (0.100)	+ 20.2
vAa4	0.279 (0.204)	− 1.0	0.818 (0.461)	+ 6.9
vAPa1	0.354 (0.195)	+ 25.8	0.918 (0.105)	+ 19.9
c4fow	0.317 (0.191)	+ 12.7	0.881 (0.148)	+ 15.1
vn	0.281 (0.195)		0.765 (0.216)	

Table 2: Evaluation measures averaged over queries (including variance)

system	avp					
vAPa1	0.358	1				
vPa1	0.354	1				
vMa1	0.350	1	2			
vMc1	0.350	1	2			
vSc1	0.348	1	2	3		
vAa1	0.347	1	2	3		
vSa1	0.346	1	2	3		
vMa3	0.344	1	2	3		
vMc3	0.344	1	2	3		
vMc2	0.341	1	2	3		
vMa2	0.340	1	2	3		
vMc4	0.330	1	2	3		
vXXX	0.330	1	2	3		
vMa4	0.330	1	2	3		
c4fow	0.319	1	2	3	4	
vn	0.313		2	3	4	5
vAa4	0.309			3	4	5
vSc2	0.286				4	5
vSc3	0.286				4	5
vSa3	0.285				4	5
vSa2	0.284				4	5
vSc4	0.279					5
vSa4	0.277					5

Table 3: Equivalence classes avp

system	ap5-15			
vMa3	0.451	1		
vPa1	0.451	1		
vMc3	0.450	1		
vAPa1	0.448	1		
vMa2	0.448	1		
vMc2	0.446	1		
vMc1	0.444	1		
vSc1	0.443	1		
vMa1	0.443	1		
vAa1	0.443	1		
vSa1	0.441	1		
c4fow	0.427	1	2	
vMa4	0.427	1	2	
vMc4	0.426	1	2	
vXXX	0.420	1	2	
vn	0.388		2	3
vAa4	0.383		2	3
vSa2	0.366			3
vSc2	0.365			3
vSc3	0.363			3
vSa3	0.361			3
vSa4	0.349			3
vSc4	0.349			3

Table 4: Equivalence classes ap5-15

system	R-Recall				
vAPa1	0.354	1			
vPa1	0.348	1	2		
vMa1	0.344	1	2		
vMc1	0.343	1	2		
vAa1	0.343	1	2		
vSc1	0.343	1	2		
vSa1	0.343	1	2		
c4fow	0.317	1	2	3	
vXXX	0.316	1	2	3	
vMa2	0.313	1	2	3	
vMa3	0.312		2	3	
vMc3	0.312		2	3	
vMc2	0.312		2	3	
vMa4	0.311		2	3	
vMc4	0.310		2	3	
vn	0.281			3	4
vAa4	0.279			3	4
vSc2	0.260				4
vSa2	0.260				4
vSc3	0.256				4
vSa3	0.254				4
vSc4	0.248				4
vSa4	0.246				4

Table 5: Equivalence classes R-recall

system	r1000							
vAa1	0.920	1						
vPa1	0.919	1						
vAPa1	0.918	1						
vSc1	0.918	1						
vSa1	0.918	1	2					
vMc1	0.914	1	2	3				
vMa1	0.914	1	2	3				
c4fow	0.881	1	2	3	4			
vMa2	0.879	1	2	3	4			
vMa3	0.875	1	2	3	4			
vMc3	0.875	1	2	3	4			
vMc2	0.873	1	2	3	4			
vXXX	0.855	1	2	3	4			
vMa4	0.850		2	3	4	5		
vMc4	0.849			3	4	5		
vAa4	0.818				4	5	6	
vSa2	0.783					5	6	7
vSc2	0.778						6	7
vn	0.765						6	7
vSc3	0.760						6	7
vSa3	0.758						6	7
vSa4	0.746							7
vSc4	0.746							7

Table 6: Equivalence classes recall1000

same equivalence class (2) as vn). Results also show that our initial attempts to improve precision by using a subset of PP post-modifiers to create new compounds (vS.. versions) were not successful. These versions are all in equivalence classes which include vn. Compared to the vM.. versions which only normalize original compounds, average precision even decreases, although in most cases the difference is not significant.

If we look at the results in more detail we see that the distinction head-modifier is not relevant for compound splitting. Versions v..1 which add all subparts of a complex term to the index usually outperform the other versions (versions v..2/3/4). If we only consider the first 15 documents retrieved (ap5-15) then versions vM.2 and vM.3 show a slight advantage over vM.1. However, this difference is not statistically significant. We also see that the two strategies for handling ambiguous structures (versions v.a. and v.c.) are equivalent. It may be that our corpus is too small to render sufficient data for the corpus-based approach. It may also be that the default strategy simply works well for our data.

In table 7 some statistics for versions c4fow and vSa1 are given. The figures show that although the number of compounds found by c4fow in greatly exceeds the number of compounds found by the syntactic version, the percentage of relevant combinations (actually found in relevant articles) is higher for the syntactic version. We concluded that the compound generation strategy employed by the vS.. versions was too restricted and should be extended to include other head-modifier pairs. We experimented with several extensions. We implemented a version which instead of a subset of PP-modifiers uses all PP-modifiers for term pair generation (version vPa1). Besides this version we also developed a version which adds noun-adjective head-modifier pairs to the index (vAa1). Version vAPa1 combines these two strategies.

version	number of compounds	relevant compounds	% relevant
c4fow	147	35	20.5
vSa1	46	18	39.1

Table 7: Relevant compounds found in queries by c4fow vs. vSa1

If we look at the results for these versions we see that version vPa1, the version which adds noun-noun pairs from all PP post-modifiers, improves precision compared to the baseline (vn). In fact, there is a statistically significant difference between this version and vn for all 4 evaluation measures. Version vAa1, the version which adds noun-adjective pairs is slightly worse than vPa1 if we look at precision but the noun-adjective pairs seem to have a positive effect on recall (see recall1000). If we combine the two types of head-modifier pairs (version vAPa1) we get the best overall results.

We may conclude that adding head-modifier pairs to the index can improve retrieval performance, but only if all constituents are also added as separate index terms. Although c4fow, the version which does not use any syntactic information, performs fairly well, especially when we look at recall, we are able to improve results even further by adding syntactic information.

6 Conclusions and future work

The results of our experiments have shown that it is possible to improve retrieval quality for Dutch texts significantly by using syntactic information to create complex index terms. Without using syntactic information we were already able to improve recall by up to 15%, but by adding syntactic information we are not only able to improve recall even further (up to 25%) but we are also able to improve precision as well (up to 16%), provided that all subparts of the complex terms are also added to the index separately. For the experiments described above we used a standard *tf.idf* term weighting scheme which does not differentiate between simple and complex index terms. Since term re-weighting schemes have proven to be successful in previous UPLIFT experiments, we intend to investigate the effect of alternative weighting strategies in the future. We also plan to adapt our strategy to English texts and investigate cross-language retrieval.

Acknowledgements

The research described in this paper was funded by Philips Research, TNO-TPD, the Utrecht Institute of Linguistics OTS, the NBBI, the Dutch Ministry of Education and Science and the Dutch Ministry of Economics. The authors would like to thank Herbert Ruessink and Dirk Heylen for their contribution to this work and the reviewers for their helpful comments on a draft of this paper.

References

Aalbersberg, I. J., E. Brandsma, and M. Corthout (1991). Full text document retrieval: from theory to applications. In G. Kempen and W. de Vroomen (Eds.), *Informatiewetenschap 1991. Wetenschappelijke bijdragen aan de eerste STINFON-Conferentie.*

Armstrong, S., P. Bouillon, and G. Robert (1995). Tools for part-of-speech tagging. Multext project report, ISSCO, Geneva.

Baayen, R. H., R. Piepenbrock, and H. van Rijn (Eds.) (1993). *The CELEX Lexical Database (CD-ROM).* University of Pennsylvania, Philadelphia (PA): Linguistic Data Consortium.

Evans, D. A. and C. Zhai (1996). Noun-phrase analysis in unrestricted text for information retrieval. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL96)*, pp. 17–24.

Fagan, J. L. (1987). *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and non-Syntactic Methods.* Ph. D. thesis, Cornell University, Ithaca NY, CS Department technical report 87-868.

Geerts, G., W. Haeseryn, J. de Rooij, and M. C. van der Toorn (Eds.) (1984). *Algemene Nederlandse Spraakkunst.* Groningen: Wolters Noordhoff.

Harman, D. (1993). Overview of the first text retrieval conference (TREC-1). In D. Harman (Ed.), *The First Text REtrieval Conference (TREC-1)*, pp. 1–20. National Institute for Standards and Technology. Special Publication 500-207.

Hays, W. L. (1978). *Statistics for the Social Sciences*. London: Holt, Rinehart and Winston.

Hull, D. A., G. Grefenstette, B. M. Schulze, E. Gaussier, H. Schütze, and J. O. Pedersen (1997). Xerox TREC-5 site report: Routing, filtering, NLP and Spanish tracks. In D. Harman (Ed.), *The Fifth Text REtrieval Conference (TREC-5)*. to appear (see http://www-nlpir.nist.gov/TREC/).

Kraaij, W. and R. Pohlmann (1996). Viewing stemming as recall enhancement. In H.-P. Frei, D. Harman, P. Schauble, and R. Wilkinson (Eds.), *Proceedings of ACM-SIGIR96*, pp. 40–48.

Salton, G. (1989). *Automatic Text Processing - The Transformation, Analysis, and Retrieval of Information by Computer*. Reading (MA): Addison-Wesley Publishing Company.

Salton, G., C. Buckley, and M. Smith (1990). On the application of syntactic methodologies in automatic text analysis. *Information Processing & Management 26*(1), 73–92.

Smeaton, A. F., R. O'Donnell, and F. Kelledy (1995). Indexing structures derived from syntax in TREC-3: System description. In D. Harman (Ed.), *Overview of The Third Text REtrieval Conference (TREC-3)*, pp. 55–63. National Institute for Standards and Technology. Special Publication 500-225.

Strzalkowski, T. (1995). Natural language information retrieval. *Information Processing & Management 31*(3), 397–417.

Strzalkowski, T. and J. Perez Carballo (1996). Natural language information retrieval: TREC-4 report. In D. Harman (Ed.), *The Fourth Text REtrieval Conference (TREC-4)*. National Institute for Standards and Technology. Special Publication 500-236.

Tague-Sutcliffe, J. (1995a). *Measuring Information. An Information Services Perspective*. San Diego (CA): Academic Press.

Tague-Sutcliffe, J. (1995b). A statistical analysis of the TREC-3 data. In D. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)*, pp. 385–398. National Institute for Standards and Technology. Special Publication 500-225.

ter Stal, W. (1996). *Automated Interpretation of Nominal Compounds in a Technical Domain*. Ph. D. thesis, Technische Universiteit Twente, UT Repro, Enschede.

van Surksum, J. and J. W. den Besten (1993, november). Patz'er - een patronenzoeker voor Nederlandstalige teksten. TNO-TPD/Hogeschool Enschede.

Vosse, T. G. (1994). *The Word Connection*. Ph. D. thesis, Rijksuniversiteit Leiden, Neslia Paniculata Uitgeverij, Enschede.

Appendix: results of the statistical analysis

The design chosen for the statistical analysis is a repeated measures single factor design, sometimes also referred to as randomized block design (see, for instance, Hays (1978), chapter 13). This design has the advantage that the query (or subject) effect is separated from the system effect. We know that different queries will render different results so if we separate this effect from the system effect we are able to single out the factor we are interested in. The statistical model for the randomized block design can be summarized as follows:

$$Y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}$$

Y_{ij} represents the score (e.g. average precision) for system variant i and query j, μ is the overall average score, α_i is the effect of the ith system, β_j is the effect of the jth query and ϵ is the random variation about the average.

The H_0 hypothesis which is tested by an analysis of variance (ANOVA) is:

> The averages of the observed statistic are equal for all system versions, i.e. the system effect (α) is zero.

If this hypothesis is falsified, we can conclude that at least one pair of averages differs significantly. T-tests are subsequently applied to determine which pairs of system versions really show a significant difference. Tables 8, 9, 10 and 11 present the results of the ANOVAs that were run on the data.

Source	DF	Sum of Squares	Mean Square	F val
system	22	0.6146	0.0279	3.9557
queries	35	35.5577	1.0159	143.8590
error	770	5.4378	0.0071	
total	827	41.6101		
s.e.d. (systems): 0.0198				

Table 8: ANOVA table average precision

Source	DF	Sum of Squares	Mean Square	F val
system	22	1.1607	0.0528	4.6266
queries	35	65.6220	1.8749	164.4138
error	770	8.7808	0.0114	
total	827	75.5635		
s.e.d. (systems): 0.0252				

Table 9: ANOVA table average precision at 5, 10 and 15 documents retrieved

The most important figures in the ANOVA tables are the F-values in the rightmost column, which represent the quotient of the variance in measurements which can be attributed to the effect we are interested in (Mean Square system or query) and the variance due to chance (Mean Square error). This quotient is dependent

Source	DF	Sum of Squares	Mean Square	F val
system	22	1.0440	0.0475	6.1685
queries	35	27.4588	0.7845	101.9800
error	770	5.9237	0.0077	
total	827	34.4265		
s.e.d. (systems): 0.0207				

Table 10: ANOVA table R-recall

Source	DF	Sum of Squares	Mean Square	F val
system	22	3.2825	0.1492	7.3180
queries	35	15.4449	0.4413	21.6433
error	770	15.6995	0.0204	
total	827	34.4269		
s.e.d. (systems): 0.0337				

Table 11: ANOVA table recall at 1000 documents

on the degrees of freedom (DF) of the variables in the model, i.e. number of system versions and queries − 1. Because the F values exceed $F_{.99;22,770}$[10] = 1.85, we may conclude that the system effect is significant at the 0.99 level for all ANOVAS, This means that we can reject the hypotheses that the system effects of the corresponding measures are equal to zero with a certainty of 99%. The query effect is also clearly significant for all evaluation measures. The F-values exceed $F_{.99;35,770} =$ 1.55. This justifies the choice for a randomized block design where the query effect is separated from the system effect.

Because the ANOVA shows that there are significant differences between system versions, it is necessary to do multiple pairwise comparisons to detect which specific versions are concerned. We have used T-tests to identify significant differences between specific versions. The standard error of difference (s.e.d.) values rendered by the ANOVA are used to discriminate significantly different versions in the following way:

$$\mid \bar{x}_1 - \bar{x}_2 \mid > 2 \times s.e.d.$$

The results of the T-tests are given in tables 3, 4, 5 and 6 in section 5 above.

---

[10] The standard value for significance level 1−0.01 and the degrees of freedom.

Unbounded Negative Concord in Polish: A Lexicalist HPSG Approach

Adam Przepiórkowski[*][†]
Anna Kupść[†][‡]

Abstract

In this paper, we deal with Negative Concord (NC) in Polish. We show that Polish NC is a kind of unbounded dependency construction (UDC), although it differs in many respects from the 'standard' UDCs such as, e.g., *wh*-extraction or topicalization. Our analysis of NC is coached in the theoretical framework of HPSG; more precisely, we adopt a lexicalist approach to UDCs proposed by Sag (1996a, 1996b). Moreover, we argue that Polish NC facts would be difficult to model by a purely semantic account.

1 Introduction

The aim of this paper is twofold. First, on the basis of facts rarely (if ever) considered in the linguistic literature, we argue for the Unbounded Dependency (UD) status of Negative Concord (NC) in Polish (and, by extension, possibly in other languages exhibiting NC). Secondly, we provide a formal HPSG analysis of the facts considered utilizing recent approaches to Unbounded Dependency Constructions (UDCs) advocated for, e.g., by Sag (1996a, 1996b). Our choice of linguistic formalism (HPSG) and the degree of formalization achieved make the account in principle computer-implementable.[1]

Negative Concord is infamous for its cross-linguistic diversity. Slavic NC contrasts with that of other languages described in the literature.[2] Section 2 presents the basic data of Polish NC; section 3 shows that NC is unbounded, although it differs in important respects from 'everyday' UDCs such as *wh*-extraction and topicalization; section 4 presents the lexical approach to UDCs which constitutes the

---

[*] Universität Tübingen

[†] PAS, Warsaw

[‡] Talana, UFRL, Paris

[1] See Bolc, Czuba, Kupść, Marciniak, Mykowiecka, and Przepiórkowski (1996) for a survey of computational formalisms for implementing HPSG grammars.

[2] See, e.g., Rizzi (1982), Zanuttini (1991) and Aranovich (1993) for Romance, and Labov (1972), den Besten (1986), Bayer (1990) and Haegeman and Zanuttini (1996) for Germanic. On the other hand, Progovac (1993, 1994) provides data from Serbo-Croatian (involving NI-NPIs in her terminology) which parallel those described in section 2, although she does not consider the unbounded aspect of NC.

basis of our analysis; and section 5 presents a detailed account of the facts described in preceding sections. In section 6 we briefly consider viability of a purely semantic approach and, finally, section 7 contains some concluding remarks.

2 Negative Concord in Polish

Polish shows both kinds of NC described in the literature, i.e., *negative doubling* and *negative spread* (cf. den Besten (1986), van der Wouden and Zwarts (1993)). We describe these species of NC below, and then move to show that licensing conditions on Polish *n*-words[3] differ from those on English Negative Polarity Items (NPIs; e.g., *any* and *ever*) or Italian *n*-words.

2.1 Negative Doubling

In Polish, sentential negation is expressed by the negative affix *nie*:[4]

(1) *Janek* ***nie*** *pomaga ojcu.*
John not helps father
'John doesn't help his father.'

Whenever any dependent of a verb, be it a subject (2a), an object (2b–c) or an adjunct (3), is a negative phrase (is or contains an *n*-word), the verb has to be preceded by the negation marker *nie*.

(2) a. ***Nikt*** ****(nie)*** *przyszedł.*
nobody not came
'Nobody came.'

b. *Marysia* ***niczego*** ****(nie)*** *dała Jankowi.*
Mary nothing not gave John
'Mary didn't give John anything.'

c. *Marysia* ****(nie)*** *dała* ***nikomu*** *książki.*
Mary not gave nobody book
'Mary didn't give anyone a/the book'.

(3) a. ***Nigdy*** ****(nie)*** *prosił o pomoc.*
never not asked-he about help
'He never asked for help.'

b. *Z* ***nikim*** ****(nie)*** *przechadzałem się wczoraj po Hradčanach.*
with nobody not strolled-I SELF yesterday on Hradčany
'I didn't stroll with anybody at Hradčany yesterday'.

---

[3] The term *n-word* was coined by—to the best of our knowledge—Laka (1990) and it has been used in much of subsequent literature on NC. It denotes those words (usually starting with the letter *n*) which enter the NC relation with the verbal negation marker (in case of *negative doubling*) or with each other (in case of *negative spread*).

[4] We argue (contra orthography) for the affix status of *nie* in Kupść and Przepiórkowski (1997).

Note that, unlike in, e.g., Italian, negative doubling does not depend in Polish on word order: preverbal negative phrases require verbal negation marker *nie* just as the postverbal ones do.

2.2 Negative Spread

Apart from negative doubling, Polish exhibits also negative spread. As the example below attests, the presence of multiple negative phrases within a clause results in a single negation meaning:

(4) ***Nikt*** ***nigdy nikogo*** ***niczym*** ****(nie)*** *uszczęśliwil.*
Nobody$_{nom}$ never nobody$_{gen}$ nothing$_{ins}$ not made happy
'Nobody has ever made anybody happy with anything.'

2.3 Licensing *N*-Words

Section 2.1 above showed that, in Polish, *n*-words require the presence of clausemate verbal negation, or, in other words, that *nie* licenses *n*-words. In many languages, including English and Italian, NPIs[5] can be licensed by a variety of environments, often characterized in semantic terms (e.g., Ladusaw (1979), van der Wouden and Zwarts (1993), Dowty (1994)). We show below that none of those NPI-licensing environments can license Polish *n*-words.[6]

Yes/no questions:

(5) * *Czy* ***nikt*** *dzwonil?*
Q nobody phoned
'Has anybody phoned?'

Indirect questions:

(6) * *Chcial wiedzieć, czy* ***nikt*** *dzwonil.*
wanted-he know, Q nobody phoned
'He wanted to know if anybody phoned.'

Adversative predicates:

(7) * *Wątpię, żeby* ***nikt*** *dzwonil.*
doubt-I that$_{subj}$ nobody phoned
'I doubt if anybody phoned.'

Antecedents of conditionals:

---

[5] We implicitly assume here that Polish *n*-words should be considered Negative Polarity Items, i.e., existential quantifiers which get their negative import from the licensing operators. The matter is, however, far from clear (see, e.g., discussion of Romance *n*-words in Laka (1990) and Zanuttini (1991)) but, fortunately, nothing hinges on this assumption.

[6] See, however, section 5 for another licensor of *n*-words.

(8) * *Jeżeli* **nikt** *dzwonił, to...*
if nobody phoned then
'If anybody phoned, then...'

Also relative clauses headed by universal quantifiers, comparatives,[7] *too*-constructions, etc. cannot license *n*-words in Polish.

3 Long Distance NC

3.1 Locality Restrictions

Subordinate clauses are in general boundaries for Negative Concord, e.g.:

(9) a. *Jan sądzi, że Marysia* **nikogo *(nie)** *lubi.*
John believes that$_{ind}$ Mary nobody not like
'John believes that Mary doesn't like anybody.'

b. * *Jan* **nie** *sądzi, żeby Marysia* **nikogo** *lubiła.*
John not believes that$_{subj}$ Mary nobody liked

(10) a. *Jan prosił, żeby* **niczego *(nie)** *ruszać w jego pokoju.*
John asked that nothing not touch$_{inf}$ in his room
'John asked not to touch anything in his room.'

b. * *Jan* **nie** *prosił, żeby* **niczego** *ruszać w jego pokoju.*
John not asked that nothing touch$_{inf}$ in his room

Note that *sądzi* in (9) is a typical 'neg-raising' predicate, that is, matrix negation can be understood as 'raised' subordinate negation:

(11) *Jan* **nie** *sądzi, żeby Marysia lubiła Tomka.*
John not believes that$_{subj}$ Mary liked Tom
'John doesn't believe that Mary likes Tom.'
(≈ 'John believes that Mary doesn't like Tom.')

Thus, if licensing conditions were a purely semantic matter, (9) would have to be explained. Moreover, it is not (as sometimes assumed) 'tenseness' that blocks NC: both in (9) and in (10) the subordinate clause does not have an independent tense. In (10) it is infinitival, while in (9) it is past participle required by the subjunctive complementizer (cf. Borsley and Rivero (1994)).

On the basis of the examples above we conclude that verbal projections (regardless of semantics or 'tenseness') constitute barriers for NC in Polish.[8]

---

[7] To be more precise, we should mention that the there is a class of comparatives which does license *n*-words. Accounting for this exception will be the topic of further research.

[8] As noted by an anonymous reviewer, the facts in (9)–(11) are also compatible with another explanation, i.e., that it is the complementizer that blocks NC. However, as discussed in Przepiórkowski and Kupść (1997a, 1997b), this explanation would be more difficult to reconcile with the behaviour of NC is complex predicates.

3.2 NPs and PPs

Note first that although *n*-words *niczyj* 'no one's' and *żaden* 'none' are not direct arguments of the verb, they still imply its negation, cf. (12):

(12) * *(**Nie**) chciałem **żadnej** książki.*
not wanted-I none book
'I didn't want any book.'

Although this behaviour could be attributed to the special status of determiners by assuming a DP analysis of noun phrases or by arguing that they 'agree' with N̄ with respect to 'negative polarity', no such explanation can be reasonably put forward to account for examples such as (13) below.

(13) *Moje stopy *(**nie**) tolerują butów z **niczego**.*
my feet not tolerate shoes from nothing
'My feet can't stand shoes made of anything.'

Moreover, there does not seem to be any constraint on the distance of Negative Concord: in (14a), NC takes place across 6 NP and PP boundaries, while in (14b), it crosses 8 such boundaries.

(14) a. **(**Nie**) lubię smaku konfitur z owoców z **niczyjego** ogrodu,*
not like-I taste of preserves from fruits from nobody's garden,
oprócz własnego.
apart my own
'I don't like the taste of preserves made of fruit from anybody's garden, apart from (these made of fruit from) my own.'

b. *Gazety z plotkami o żonach władców państw **żadnego***
Newspapers with rumours about wives of rulers of countries of none
*kontynentu *(**nie**) są tak interesujące, jak te z plotkami o*
continent not are so interesting as those with rumours about
żonach władców państw afrykańskich.
wives of rulers of countries African
'No newspapers with gossip about wives of rulers of countries of any continent are so interesting, as these containing gossip about wives of rulers of African countries.'

3.3 Summary

Thus, we conclude that Polish Negative Concord is a species of UDCs, although it differs from such well-known UDCs as *wh*-extraction or topicalization in many important respects. First, it is unbounded in the sense that it can work across arbitrarily many NP and PP projections, unlike, e.g., English *wh*-extraction (cf. * *Whose do you like mothers?*). Moreover, subordinate clauses constitute barriers to NC, regardless of whether they are tensed. Additionally, there is no gap whose filler should be found; the dependency is rather introduced lexically by *n*-words.

Finally, unlike the so-called 'strong' UDCs (cf. Pollard and Sag (1994)), there is no overtly realized element corresponding to the dependency.[9]

4 Lexical Approach to UDCs

In what follows, we will build on the lexical approach to unbounded dependency constructions (UDCs) of Sag (1996a, 1996b). The main idea of this approach is that normally words inherit SLASH values of their arguments by simply amalgamating them, i.e., they satisfy the principle of 'Lexical Amalgamation of SLASH':

(15) Lexical Amalgamation of SLASH:

$$\begin{bmatrix} \text{ARG-S} \ \langle [\text{SLASH} \ \boxed{1}], \ldots, [\text{SLASH} \ \boxed{n}] \rangle \\ \text{SLASH} \ \boxed{1} \uplus \ldots \uplus \boxed{n} \end{bmatrix}$$

Moreover, 'SLASH Inheritance Principle' takes care of percolating the value of SLASH from such lexical entries to their maximal projections.[10]

(16) SLASH Inheritance Principle (an approximation):

$$\textit{hd-nexus-ph} \rightarrow \begin{bmatrix} \text{NONLOCAL}|\text{SLASH} \ \boxed{1} \\ \text{HEAD-DTR}|\text{NONLOCAL}|\text{SLASH} \ \boxed{1} \end{bmatrix}$$

One advantage of this approach over any purely syntactic treatment of UDCs is that it allows to easily account for the cases in which an unbounded constituent is discharged lexically. The classical example are *easy*-adjectives, e.g.:

(17) *I am easy to please* ___.

In sentences such as (17), the missing object of the lower verb is nowhere to be found; the nominative subject, *I*, cannot be the filler for the missing object, although it is understood as coreferential with it. In the framework sketched above, this can be easily accounted for by positing that *easy*-adjectives are exceptional in that they do not satisfy the principle of Lexical Amalgamation of SLASH, but rather remove one element from the sum of SLASH values of their arguments and coindex it with their subject.[11]

5 The Analysis

There are many reasons for applying Sag's lexical approach to unbounded Negative Concord in Polish. First of all, the 'negation requirement' is introduced lexically

---

[9] The marker *nie* can hardly be considered one: multiple clausemate *n*-words trigger just one verbal negation (although this could be explained by postulating obligatory haplology of *nie*) and there is another element which can license *n*-words, namely the preposition *bez* (see section 5).

[10] SLASH Inheritance is in work only for certain kinds of phrases, namely *head-nexus-phrases*, in order to exclude items that bind SLASH lexically, cf. Sag (1996a).

[11] Examples such as (17) can be also accounted for assuming the approach to UDCs of Pollard and Sag (1994). However, this approach fails on attributive uses of *easy*-phrases as in (i) below:

(i) *An easy to please man came yesterday.*

See Sag (1996a, 1996b) for details.

(by *n*-words such as *nikt*, *nigdy* and *żaden*). Secondly (and more importantly), 'negation requirement' is discharged lexically, by morphologically negated verbs. Finally, there is an interesting lexical exception to the generalization that prepositions always let the negation requirement percolate higher up: the preposition *bez* 'without' binds negation.

(18) a. *Zaczął bez* ***żadnych*** *wstępów*.
started-he without none introductions
'He started straight away.'

b. *Został bez* ***niczego***.
stayed-he without nothing
'He was left broke.'

This exception would be awkward to model in the syntax.[12]

In the remainder of this section, we formalize the observations made above.

5.1 Nonlocal Attribute NEGATIVE-CONCORD

In order to account for these facts, we introduce a non-local attribute responsible for Negative Concord, NEG-CONC. Since it does not matter what kind of negative elements initiate the negation, nor does it matter from exactly how many arguments negation percolates, we will assume that the only values of this attribute are '+' and '–'.

(19)
$$\begin{bmatrix} \textit{nonlocal} \\ \text{NEG-CONC boolean} \\ \ldots \end{bmatrix}$$

5.2 Introducing Negation Requirement

The negation requirement is always introduced by negative elements. This is done lexically by positing that such elements have the value of NEG-CONC set to '+' in the lexicon.

(20)
$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \text{nikt} \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOC}|\text{CAT}|\text{HEAD} & \begin{bmatrix} \textit{noun} \\ \text{CASE nom} \end{bmatrix} \\ \text{NONLOC}|\text{NEG-CONC} & + \end{bmatrix} \end{bmatrix}$$

5.3 Cancellation

The lexical items which cancel negation percolation have '–' set up in the lexicon as the value of their NEG-CONC, e.g.:[13]

---

[12] See, e.g., Progovac (1993), which assumes that *without*-headed prepositional phrases project to clauses.

[13] Constraint (21) should be ideally understood as a constraint on the lexicon saying that all verbal lexical entries have to be NEG-CONC–. Alas, this cannot be expressed in pure HPSG, so we model this generalization by leaving the value of NEG-CONC underspecified on lexical entries and positing constraint (21), whose role is to resolve this value to '–'.

(21)
$$\begin{bmatrix} \textit{word} & \\ \text{SYNSEM|LOC|CAT|HEAD} & \begin{bmatrix} \textit{verb} \\ \text{NEG} \; + \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYNSEM|NONLOC|NEG-CONC} \; - \end{bmatrix}$$

(22)
$$\begin{bmatrix} \textit{word} & \\ \text{PHON} \; \langle \text{bez} \rangle & \\ \text{SYNSEM} & \begin{bmatrix} \text{LOC|CAT|HEAD} & \begin{bmatrix} \textit{prep} \\ \text{PFORM bez} \end{bmatrix} \\ \text{NONLOC|NEG-CONC} \; - & \end{bmatrix} \end{bmatrix}$$

Note that this specification correctly models both the cases in which none of the arguments of a cancelling item is an n-word, and those in which there are some n-words among the arguments. In the former case, there is simply no 'negation requirement' to percolate higher up, so the NEG-CONC value should be '−'. In the latter case, the 'negation requirement' is cancelled, hence it should not percolate up, so the NEG-CONC value should be again '−'.

5.4 Percolation

Following Sag's approach to UDCs (see section 4 above), we assume that 'negation percolation' is happening in two steps. First, the percolating item 'amalgamates' the information on 'negation requirement' from its arguments; then this information is transmitted along the head projection path.

5.4.1 Negation Amalgamation

The lexical items which allow percolation of negation specify the value of their NEG-CONC as '+' if at least one of their arguments is NEG-CONC+, and as '−' otherwise. This is analogous to Sag's Lexical Amalgamation of SLASH.[14]

(23) Lexical Amalgamation of NEG-CONC:[15]

$$\begin{bmatrix} \textit{word} & \\ \text{SYNSEM|LOC|CAT} & \begin{bmatrix} \text{HEAD noun} \vee \text{prep} \\ \text{ARG-S} \; \boxed{1} \, \text{list(synsem)} \end{bmatrix} \end{bmatrix} \rightarrow$$
$$\begin{bmatrix} \text{SYNSEM|NONLOC|NEG-CONC} \; \boxed{2} \end{bmatrix} \wedge \text{sum\_neg}(\boxed{1}, \boxed{2})$$

In the constraint above, sum_neg/2 denotes the relation which holds between a list and a boolean value only if either there is an $\begin{bmatrix} \text{NONLOC|NEG-CONC} \; + \end{bmatrix}$ element in the list and the boolean value is '+', or if there is no such element and the boolean value is '−':

---

[14] We assume that all dependents, including modifiers, are on ARG-S of a verb, compare Miller (1992), van Noord and Bouma (1994), Manning, Sag, and Iida (1997) and Przepiórkowski (1997b, 1997a).

[15] Again, this constraint is somewhat sloppy. It should be understood as a default constraint on nominal and prepositional lexical entries (or, otherwise, particular lexical entries would have to be idiosyncratically marked as amalgamating items); it can be overridden by n-words (e.g., (20)) and the preposition *bez* (22). This could be formalized via mechanisms of the kind postulated by Sag and Miller (1997) and Abeillé, Godard, and Sag (1997) (defaults and hierarchical lexicon). Unfortunately, we are not aware of explicit formalizations of these mechanisms within HPSG.

$$\begin{array}{l}
\textbf{sum\_neg}(\langle\rangle, -).\\
\textbf{sum\_neg}(\langle \left[\textsc{nonloc}|\textsc{neg-conc}\ +\right] \rangle \oplus \textbf{list}, +).\\
\textbf{sum\_neg}(\langle \neg \left[\textsc{nonloc}|\textsc{neg-conc}\ +\right] \rangle \oplus \boxed{1}, \boxed{2}) :-\\
\qquad \textbf{sum\_neg}(\boxed{1}, \boxed{2}).
\end{array}$$

5.4.2 Negation Inheritance Constraint

The second step ensures percolation of the NEG-CONC value along the head projection from a lexical item to its maximal projection. This is done with the help of Negation Inheritance Constraint (NIC), a constraint analogous to the SLASH Inheritance Principle (cf. (16) above).

(24) Negation Inheritance Constraint (NIC):

$$\begin{bmatrix} \textit{phrase} \\ \text{DTRS headed-struc} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYNSEM|NONLOCAL|NEG-CONC } \boxed{1} \\ \text{DTRS|HEAD-DTR|SYNSEM|NONLOCAL|NEG-CONC } \boxed{1} \end{bmatrix}$$

Note that, unlike in (16), there is no need to exclude phrases overtly realizing a missing constituent from the Negation Inheritance Constraint (because there is no missing constituent in this UDC), so (24) is a constraint on all headed phrases.

5.5 Islands

Islands for NC (non-negated verbs) can be characterized by two features: they do not allow any arguments to introduce the 'negation requirement'; and they themselves do not introduce the 'negation requirement'. In terms of the analysis above, this means that lexical entries which create islands for NC require that all their arguments be NEG-CONC− and that they are NEG-CONC− themselves. The second condition amounts to saying that island-creating items belong to the class of 'cancelling items'. Interestingly, the first condition then amounts to saying that these items also have to belong to the class of 'percolating items'. (That is because under the assumption that they are NEG-CONC− and that the value of NEG-CONC can be only either '+' or '−', the statements "all their arguments are NEG-CONC−" and "some of their arguments can be NEG-CONC+ only if they are NEG-CONC+" (which they are not!) are logically equivalent.)

Thus, in order to account for islands for NC, all there is to do is to include island-creating items (non-negated verbs) in the antecedents of constraints (21) and (23):[16]

(21′)

$$\begin{bmatrix} \textit{word} \\ \text{SYNSEM|LOC|CAT|HEAD verb} \end{bmatrix} \rightarrow \left[\text{SYNSEM|NONLOC|NEG-CONC } - \right]$$

(23′) Lexical Amalgamation of NEG-CONC:

$$\begin{bmatrix} \textit{word} \\ \text{SYNSEM|L|CAT} & \begin{bmatrix} \text{HEAD noun} \vee \begin{bmatrix} \textit{prep} \\ \text{PFORM } \neg\text{bez} \end{bmatrix} \vee \begin{bmatrix} \textit{verb} \\ \text{NEG } - \end{bmatrix} \\ \text{ARG-S } \boxed{1}\,\text{list(synsem)} \end{bmatrix} \end{bmatrix} \rightarrow$$

---

[16] See footnotes 13 and 15.

$$\Big[\,\text{SYNSEM|NONLOC|NEG-CONC}\ \boxed{2}\,\Big] \land \text{sum\_neg}(\boxed{1},\boxed{2})$$

5.6 An Example

Lexical entries such as (20) and (22), together with constraints (21′), (23′) and (24) correctly account for the negation data in (1)–(10), (12)–(14) and (18). We will further illustrate this analysis with example (25).

(25) *Janek* ***nigdy*** ****(nie)*** *czytał* ***żadnych*** *książek.*
John never not read none books
'John has never read any books.'

- There are three dependents of the verb: the subject *Janek*, the adverbial modifier *nigdy*, and the object *żadnych książek*;
- the object's head is the word *książek*, its only dependent is the negative element *żadnych*, which is NEG-CONC+, so *książek*, according to (23′), is also NEG-CONC+;
- through NIC (cf. (24)), NEG-CONC+ percolates to the maximal projection of *książek*, i.e., the phrase *żadnych książek* is NEG-CONC+;
- *nigdy* is specified in the lexicon as NEG-CONC+;
- the subject's head is *Janek*, it is a noun with no dependents, so, according to (23′), it is NEG-CONC−;
- according to NIC, NEG-CONC− percolates to the maximal projection of *Janek*;
- let us first consider ungrammatical (25) with no overt negation on the verb: *czytał* is a non-negated verb, so:
  - (21′) applies, hence *czytał* is NEG-CONC−;
  - (23′) applies, some of the dependents of the verb are NEG-CONC+, so the verb is also NEG-CONC+;
  - a contradiction, so the sentence with non-negated verb is ungrammatical;
- On the other hand, (25) with negation is correct: *nie czytał* is a negated verb, so:
  - (21′) applies, so *nie czytał* is NEG-CONC−;
  - (23′) does not apply, so no contradiction ensues;
  - NIC applies, NEG-CONC− is projected to the top of the clause;
  - as a result, we get a NEG-CONC− sentence, i.e., a sentence with no undischarged 'negation requirement.'

6 A Purely Semantic Account?

It has been often proposed that NC is an essentially semantic phenomenon (e.g., van der Wouden and Zwarts (1993), Progovac (1993), Acquaviva (1995)). We are sympathetic with the view that at least partially semantic solution should be sought (e.g., to explain the fact that the preposition *without* licenses NC in many languages). However, the analysis of Polish NC has to be to a large extent syntactic in view of the arguments presented below.

Neg-Raising 'Neg-raising' (scope of negation) does not license *n*-words in Polish (unlike in some other languages). As (26a) (=(11)) shows, negating the matrix verb *sądzi* 'believes' may have the 'neg-raising' effect. However, as shown in (26b) (=(9b)), this does not suffice to license the downstairs *n*-word.

(26) a. *Jan **nie** sądzi, żeby Marysia lubiła Tomka.*
John not believes that$_{subj}$ Mary like$_{pst\text{-}part}$ Tom
'John believes that Mary doesn't like Tom.' (possible reading)

b. * *Jan **nie** sądzi, żeby Marysia **nikogo** lubiła.*
John not believes that$_{subj}$ Mary nobody like$_{pst\text{-}part}$
'John believes that Mary doesn't like anybody.' (putatively)

Verb Clusters As discussed in (Przepiórkowski and Kupść 1997b), an *n*-word dependent of the lowest verb in a verb cluster triggers *nie* on any of the verbs in the cluster:

(27) *Janek *(**nie**) chce pójść do **żadnego** kina.*
John not wants go$_{inf}$ to none cinema
'John doesn't want to go to any cinema.'

On the other hand, the presence of an intervening complementizer disallows this:

(28) * *Janek **nie** chce, żeby pójść do **żadnego** kina.*
John not wants that$_{subj}$ go$_{inf}$ to none cinema
'John doesn't want one to go to any cinema.' (putatively)

It is difficult to see what semantic factors could explain this contrast.

Gerunds As mentioned in (Przepiórkowski and Kupść 1997a), gerunds behave in a different way than verbs do, i.e., they optionally let the negation requirement percolate higher up:

(29) a. ? *Napisanie poprawnie **żadnego** dyktanda *(**nie**) pomoże mu w*
writing correctly$_{adv}$ no dictation not will help him in
wygraniu konkursu.
winning competition

'Completing correctly no dictation exercise will help him to win the competition.'

b. *Poprawne napisanie* ***żadnego*** *dyktanda* ****(nie)*** *pomoże mu w wygraniu*
correct$_{adj}$ writing no dictation not will help him in winning
konkursu.
competition
'Correct completion of no dictation exercise will help him to win the competition.'

Although for some speakers there is a slight difference in grammaticality between verbal gerunds (whose CONTENT is argued by (Malouf 1996) to be the same as that of corresponding verbs) and nominal gerunds, cf. (29a) vs. (29b), a much stronger contrast is to be expected if NC is a purely semantic phenomenon.

Cross-linguistic Variation There is a good deal of cross-linguistic variation in NC (cf., e.g., the works on Romance and Germanic cited in this paper) which, as far as we can see, cannot be explained on purely semantic grounds.

7 Conclusions

The main aims of this paper were to show that Negative Concord in Polish is a species of Unbounded Dependency Constructions and to provide a formal analysis of this phenomenon. Our analysis is hosted in HPSG, more specifically, it utilizes the lexical approach to UDCs of Sag (1996a, 1996b). This way we were able to account for a kind of unbounded dependency without missing constituents.

It should be noted that the idea that NC is in some sense an unbounded dependency is not unique to our proposal (although the set of data supporting this conclusion is). In much of GB-based work on NC in various languages, various island constraints similar to those of other kinds of UDCs were noted, cf., e.g., Rizzi (1982), Bayer (1990), Zanuttini (1991), Haegeman and Zanuttini (1996), Progovac (1993).[17] However, since there is no overt movement in NC, the only way to deal with those observations was to assume movement at Logical Form. In HPSG, on the other hand, although only a single level of representation is available, various kinds of UDCs can be accounted for with the help of the same mechanism, namely amalgamation and inheritance of non-local features. However, since different non-local features are involved in NC and, say, *wh*-extraction, any differences between these two kinds of UDCs can be easily parameterized.

To put our results in the broader perspective, it is useful to compare them to the approach advocated by Progovac (1988, 1993, 1994). On her account, NC is a close relative to binding insofar that negative polarity items have to be locally bound by a negative operator (e.g., sentential negation marker) while positive polarity items have to be locally free. Contrary to appearances, we consider our 'unbounded' approach to NC compatible with Progovac's 'binding' approach to negative polarity. For example, it is striking that in Polish anaphora binding seems to be unbounded in the same way as NC, i.e., it can cross NP and PP projections (even if an accessible subject of an NP is available), compare (14a) above with (30) below:

---

[17] See also (Recourcé 1995) for an HPSG analysis of French NC as a kind of UDC.

(30) *Janek lubi smak konfitur z owoców tylko ze swojego ogrodu.*
John likes taste of preserves from fruits only from ANA POSS garden
'John likes the taste of preserves made of fruit only from his own garden.'

These similarities between NC and binding certainly deserve further research.

References

Abeillé, A., D. Godard, and I. A. Sag (1997). Two kinds of composition in French complex predicates. Version of June 29, 1997. To appear in Erhard Hinrichs, Andreas Kathol, and Tsuneko Nakazawa, eds., *Complex Predicates in Nonderivational Syntax*. New York: Academic Press.

Acquaviva, P. (1995). The logical form of negative concord. In *Résumés des Communications du Colloque de Syntaxe et Sémantique de Paris*. Université Paris 7.

Aranovich, R. (1993). Negative concord in Spanish and *in-situ* licensing. In E. Duncan, D. Farkas, and P. Spaelti (Eds.), *Proceedings of Twelfth West Coast Conference on Formal Linguistics*.

Bayer, J. (1990). What Bavarian negative concord reveals about the syntactic structure of German. In J. Mascaro and M. Nespor (Eds.), *Grammar in Progress*, pp. 13–24. Dordrecht: Foris Publications.

Bolc, L., K. Czuba, A. Kupść, M. Marciniak, A. Mykowiecka, and A. Przepiórkowski (1996). A survey of systems for implementing HPSG grammars. Technical Report 814, Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland.

Borsley, R. D. and M. L. Rivero (1994). Clitic auxiliaries and incorporation in Polish. *Natural Language and Linguistic Theory 12*, 373–422.

den Besten, H. (1986). Double negation and the genesis of Afrikaans. In P. Muysken and N. Smith (Eds.), *Substrata versus Universals in Creole Languages. Papers from the Amsterdam Creole Workshop*, Amsterdam/Philadelphia, pp. 185–230. Benjamins.

Dowty, D. (1994). The role of negative polarity and concord marking in natural language reasoning. In *Proceedings of the Fourth Conference on Semantics and Linguistic Theory*.

Haegeman, L. and R. Zanuttini (1996). Negative concord in West Flemish. In A. Belletti and L. Rizzi (Eds.), *Parameters and Functional Heads*, Oxford Studies in Comparative Syntax, Chapter 4, pp. 117–179. New York: Oxford University Press.

Kupść, A. and A. Przepiórkowski (1997, January). Verbal negation in Polish: Syntax or morphology? Unpublished manuscript.

Labov, W. (1972). Negative attraction and negative concord in English grammar. *Language 48*, 773–818.

Ladusaw, W. A. (1979). *Polarity sensitivity as inherent scope relations.* PhD dissertation, University of Texas at Austin, Austin, Texas.

Laka, I. (1990). *Negation in Syntax: On the Nature of Functional Categories and Projections.* PhD dissertation, MIT, Cambridge, Mass.

Malouf, R. (1996). Mixed categories in hpsg. Paper presented at Third International Conference on HPSG, Marseille.

Manning, C. D., I. A. Sag, and M. Iida (1997, March). The lexical integrity of Japanese causatives. To appear in G. Green and R. Levine (eds.), *Readings in HPSG*, Cambridge University Press. Version of March 5, 1997.

Miller, P. (1992). *Clitics and Constituents in Phrase Structure Grammar.* New York: Garland.

Pollard, C. and I. A. Sag (1994). *Head-driven Phrase Structure Grammar.* Chicago University Press.

Progovac, L. (1988). *A Binding Approach to Polarity Sensitivity.* PhD dissertation, University of Southern California, Los Angeles.

Progovac, L. (1993). Negative polarity: Entailment and binding. *Linguistics and Philosophy 16*, 149–180.

Progovac, L. (1994). *Negative and Positive Polarity.* Cambridge: Cambridge University Press.

Przepiórkowski, A. (1997a). On case assignment and "adjuncts as complements". Unpublished manuscript. To be submitted to the HPSG volume in "Studies in Constraint-Based Lexicalism".

Przepiórkowski, A. (1997b). Quantifiers, adjuncts as complements, and scope ambiguities. Draft of July 3, 1997.

Przepiórkowski, A. and A. Kupść (1997a). Negative concord in Polish. Technical Report 828, Institute of Computer Science, Polish Academy of Sciences.

Przepiórkowski, A. and A. Kupść (1997b). Verbal negation and complex predicate formation in Polish. In *Proceedings of the 1997 Conference of the Texas Linguistics Society on the Syntax and Semantics of Predication*, Austin. To appear.

Recourcé, G. (1995, March). *L'association négative en français. Etude linguistique et formelle de la particule ne.* Ph. D. thesis, Université Paris 7, UFRL.

Rizzi, L. (1982). *Issues in Italian Syntax*, Chapter IV: Negation, *Wh*-movement and the null subject parameter, pp. 117–184. Dordrecht: Foris Publications.

Sag, I. A. (1996a). English relative clause constructions. To appear in *Journal of Linguistics*.

Sag, I. A. (1996b). Head-driven extraction. In progress. Version of April 16, 1996.

Sag, I. A. and P. H. Miller (1997). French clitic movement without clitics or movement. To appear in *Natural Language and Linguistic Theory*.

van der Wouden, T. and F. Zwarts (1993). A semantic analysis of negative concord. In U. Lahiri and A. Wyner (Eds.), *Proceedings of the Third Conference on Semantics and Linguistic Theory*, Ithaca. Cornell University, Linguistics Department.

van Noord, G. and G. Bouma (1994). Adjuncts and the processing of lexical rules. In *Fifteenth International Conference on Computational Linguistics (COLING '94)*, Kyoto, Japan.

Zanuttini, R. (1991). *Syntactic Properties of Sentential Negation. A Comparative Study of Romance Languages.* PhD dissertation, University of Pennsylvania.

Some of the authors' papers are available electronically via WWW from `http://www.ipipan.waw.pl/mmgroup/papers.html`.

Information Update in Dutch Information Dialogues

Mieke Rats[*][†]

Abstract

In this paper, a framework is developed for the study of information update in naturally occurring information dialogues, that takes into account the information structure of the individual utterances. It is based on the dialogue theory of Bunt(1994,1995), the topic management work of Rats (1996), and the information packaging theory of Vallduví (1990). The framework is tested on a corpus of 111 telephone conversations recorded at the information service of Schiphol. The results are promising which gives us the hope that it may serve as the point of departure for a study of information packaging in other naturally occurring conversations as well.

Introduction

In this paper, we will describe a theoretical framework for the study of information update in naturally occurring conversations. Our description will use the information packaging ideas for information update of Vallduví (1990). Until now, information packaging was studied for isolated utterances or isolated utterance pairs, often thought up rather than empirically observed. We now want to apply them to spontaneous human conversation, that has the goal to exchange factual information about a specific domain.

Since Vallduví's work is confined to the analysis of isolated sentences alone, we will have to make the theoretical framework suitable for dialogue analysis. To reach this end, we will integrate the information packaging ideas for information update in the dialogue theory developed by Bunt(1994,1995). We will also use the work of Rats(1994,1995a,1995b,1996), who has extended the theory of Bunt with the notions of *topic* and *comment*. This framework will be refined with the information packaging notions *focus* and *tail*. Our set up will be illustrated by dialogue fragments taken from a corpus of 111 telephone conversations recorded at the information service of Schiphol.

---

[*] This paper reports about research done at the Human Communication Research Center, University of Edinburgh, Scotland and the Institute of Language Technology and Artificial Intelligence, Tilburg University. I am greatly indebted to Harry Bunt for his fruitful comments on the paper.

[†] At the moment, the author works at Knowledge-based systems, Technical Informatics, TU Delft

The resulting theory will not only be suitable for the study of information packaging in dialogue, it will also provide a more structured description of the dialogue partner's "mental state" Vallduví talks about when he wants to explain the speaker's choice for certain information packagings. Our analysis will show that the speaker's choice for certain information packagings will initially be determined by the introduction of the topic of the conversation. Once the topic is set, his choices will presuppose the context built up so far and depend on the way in which he wants to change it.

The outline of the paper is as follows. Section 1 will describe the information packaging ideas, that we want to integrate. Section 2 and 3 will give an overview of the work of Bunt and Rats and the relation between information packaging ideas and the theory of topic management of Rats will be explained. Section 4 will show the incorporation of the notions *focus* and *tail*. The paper will end with some proposals for further research.

1 Information packaging

Information packaging theorists are interested in the way in which people present the information content of their utterances (Chafe (1976), Vallduví(1990,1994a, 1994b,1994c), Vallduví and Engdahl (1994)). The following two utterances, for instance, show different packagings of the same information:

> The KL 627 will arrive at FIVE PAST TWO
> The KL 627 will ARRIVE at five past two

In the first utterance, the speaker has put an accent on "five past two", while in the second utterance the accent is placed on "arrive".

Information packaging does not only concern sentence accent placement, but also word order and the use of special syntactic structures. The following example stems from Rats (1996).

> We don't get passenger lists.
> Passenger lists we don't get.

The two utterances contain essentially the same information. But the information is presented in different ways. The first sentence exhibits unmarked word order, while the second contains a topicalization construction.

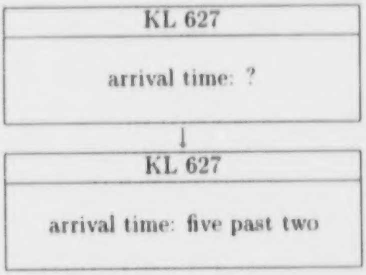
According to the information packaging literature, different linguistic choices reflect different assumptions of the speaker about the information state of the listener. A speaker will construct his utterance in such a way that the information he wants to communicate will be most easily integrated in the presumed information state of the listener. If he considers his message as completely new, for instance, he will present it as completely new. But if he thinks the new information can be attached to an information structure already available in the listener's consciousness, he will present it accordingly.

According to Vallduví, different packagings reflect different update instructions. Each instruction indicates what part of the utterance constitutes the information that has to be updated, according to the speaker's assumptions, and eventually where and how that information fits in the listener's information store. An utterance contains references to at most three informational components:

1. a link, a sentence element that refers to the locus of update,
2. a tail, a sentence element that refines the locus of update, and
3. a focus, a sentence element that points to the actual update potential.

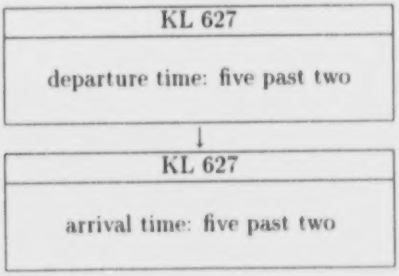
Vallduví uses Heim's file metaphor (Heim (1983)) to describe the roles of the three kinds of reference in the information update more exactly. The information store of the listener could be seen as a collection of entity-denoting file cards. On each file there are entries recording relations and attributes of the entity denoted by that file-card. The content of the file cards is updated during communication. The three informational references each play their own role in making this process more efficient. The *link* points to a specific file card, the *focus* is the information that the listener has to update on that file-card, and the *tail* specifies more exactly where the focus fits on the given file card.

Applied to the above examples, the speaker may assume the following update for the utterance *The KL 627 will arrive at FIVE PAST TWO*:



The speaker assumes that the listener doesn't know or doesn't have the right information about the exact arrival time of flight KL 627. In his knowledge store, there is a file card for flight KL 627 and the file card has a slot for the arrival time. The speaker tells him with what information he can fill this slot.

The following figure shows a possible update for the sentence *The KL 627 will ARRIVE at five past two*:



The listener is assumed to have a file card of the flight and to know about a time connected with this flight, but he has connected this time with the wrong attribute. The speaker tells him in which slot this time needs to be filled.

The utterances in these examples each have a link, a tail and a focus. But the link and the tail need not always be expressed. This may happen, for instance, in a context were both the link and the tail are already available. Since each utterance in a meaningful discourse has the intention to be informative (Chafe (1987)), only the focus is obligatory. This means that there are four possible information structures for utterances:

All-focus	in which case the speaker instructs the listener to add the information content of the whole utterance to his information state.
Link-focus	in which case the speaker instructs the listener to open a specific file-card and to add, revise, etcetera the focus on this specific file-card.
Link-tail-focus	in which case the speaker instructs the listener to open a specific file-card and a specific slot and to add, revise, etcetera the focus on this particular place.
Tail-focus	in which case the speaker presupposes that the listener has the link available and he only instructs the listener to go to a particular slot where he must add, revise, etcetera the focus.

According to Vallduví, these information structures manifest themselves in the linguistic form of the utterances. The linguistic realization varies from language to language. For English, prosody plays an important role in the structural encoding of information packaging. The structural difference between, for instance, a link-focus sentence as utterance (1) below and a link- focus-tail sentence as utterance (2) below in English is exclusively expressed by prosodic means.

(1) [$_L$ The KL 507][$_F$ will ARRIVE in time]
(2) [$_L$ The KL 507][$_F$ will ARRIVE] in time

By contrast, in Catalan syntax is the important device by which information packaging choices are expressed. It would be interesting to study the information packaging devices for Dutch. Before this can be done, however, we need to integrate the information packaging ideas for information update described in this section into a dialogue theory. For that purpose, we will use Bunt(1994,1995) and the framework developed in Rats (1996) for topic management in information dialogues. At the same time, we can see if the integration of the information packaging ideas will lead to an acceptable theory for information update in dialogue.

2 Information dialogues

The framework of Rats (1996) is based on a study of 111 naturally occurring telephone conversations, recorded at the information service of Schiphol Airport (Amsterdam International Airport). The conversations belong to the genre of *information dialogues.* Characteristic for such dialogues is that there is an information seeker who needs some information about a certain domain, and an information service that has information about that domain. In our case, the domain is the world of flights and things that have to do with flights, such as passengers, luggage etc. An example of such a dialogue is the following:

**2063

1	I:	Inlichting Schiphol	Schiphol Information
2	S:	Ja,	Yes,
3		u spreekt met de Wijl	you are speaking with de Wijl
4		Vlucht KL 550,	Flight KL 550,
5		hoe laat is die gepland?	for what time is it scheduled?
6	I:	Die wordt nu definitief verwacht om vijf voor twaalf	It is now definitely expected at five to twelve
7	S:	Vijf voor twaalf?	Five to twelve ?
8	I:	Ja hoor	Yes indeed
9	S:	Oké,	Okay,
10		bedankt	thank you
11	I:	Tot uw dienst	You're welcome
12	S:	Dag	Goodbye
13	I:	Dag	Goodbye

In this dialogue information is exchanged about flight KL 550.

The dialogues were analysed according to the Dynamic Interpretation Theory (DIT) of Bunt(1994,1995). The basic units of analysis are taken to be *utterances*, sentences or other grammatical units (words or phrases) that express one or more *dialogue acts.* Dialogue acts are defined as functional units used by the speaker to modify the dynamic context. They bring the dialogue context, which contains the information states of the two participants, from one state to an other. A dialogue act has an information content and a communicative function. The communicative function will determine how the information content of the act will be integrated into the context.

Looking at the example dialogue, it may be observed that not all utterances concern exchange of information about a topic in the task domain. We see utterances that concern various aspects of the communication at a meta-level, like introducing oneself, showing contact, greeting, and showing acceptance, gratefulness and willingness to cooperate. These aspects, which are very important for a successful and smooth information exchange, seem rather marginal with respect to topic management. Bunt (1994) has called these acts *dialogue control acts.*

For the description of topic management, we restrict ourselves to only those dialogue acts that really concern information exchange about domain topics. These are

- topic management acts
  - explicit topic introductions
  - explicit topic shifts
- informative acts
  - wh-questions and wh-answers

- yes/no-questions and yes/no-answers
- checks, confirms, and disconfirms
- alternative-questions and alternative-answers
- informs
- corrections

In the example dialogue, only utterances 4, 5, 6, 7 and 8, concern information exchange about a certain topic of the task domain.

**2063

1	I:	Inlichting Schiphol	Schiphol Information	
2	S:	Ja,	Yes,	
3		u spreekt met de Wijl	you are speaking with de Wijl	
4		Vlucht KL 550,	Flight KL 550,	Explicit topic introduct
5		hoe laat is die gepland?	for what time is it scheduled?	Wh-question
6	I:	Die wordt nu definitief verwacht om vijf voor twaalf	It is now definitely expected at five to twelve	Wh-answer
7	S:	Vijf voor twaalf?	Five to twelve ?	Check
8	I:	Ja hoor	Yes indeed	Confirm
9	S:	Oké,	Okay,	
10		bedankt	thank you	
11	I:	Tot uw dienst	You're welcome	
12	S:	Dag	Goodbye	
13	I:	Dag	Goodbye	

3 Topic management

In Rats (1996), topic management is described in an incremental way. First, the topic-comment structures of individual utterances are determined. Then, it is shown how the topic-comment structures of the individual utterances are combined to form a topic-comment structure of a dialogue fragment.

The description starts with the following definitions of topic and comment for dialogue acts (cf. Gundel(1985,1988)):

> *An entity, T, is the topic of a dialogue act, D, if D is intended to increase the addressee's knowledge about, request information about or otherwise get the addressee to act with respect to T.*
>
> *Information, C, is the comment of a dialogue act, D, if D is what is actually communicated, i.e., asserted, questioned with respect to the topic.*

We will show how these definitions work by applying them to each of the utterances of the following dialogue fragment:

**2063

		...
4		Flight KL 550,
5		for what time is it scheduled?
6	I:	It is now definitely expected at 11.55.
7	S:	11.55?
8	I:	Yes.
		...

In utterance 4, a topic is introduced: Flight KL 550. In utterance 5 information is requested about it:

> *for what time is it scheduled?*

The topic of this utterance, the entity about which the information is asked, is represented by *it*. The rest of the utterance *for what time is _ scheduled?* represents the information that is asked about it, the comment.

Utterance 6 provides the requested information:

> *It is now definitely expected at five to twelve.*

The topic of this utterance, the entity about which the information is provided is again represented by *it*. The comment, the information provided about it, is represented by the rest of the utterance *_ is now definitely expected at five to twelve.*

Utterance 7 checks this information by repeating part of utterance 6:

> *Five to twelve?*

The topic and also a large part of the comment is left out. Only part of the comment of the preceding utterance is expressed. Nevertheless, the topic this piece of information is about is still the same: Flight KL 550.

After consequent application of these definitions to the individual informative acts in the corpus, the acts could be connected with the topics functioning as links, as is illustrated by example dialogue **4379.

The analysis shows that the information exchange in the conversation is organized around one topic, topic T_1, the JU 222. Stated in another way: the topic is the connecting thread between the individual utterances in the dialogue. In fact, all dialogues in the corpus exhibit one or more of these topical lines.

We may derive from this that the function of topic management is to provide the speakers with a point of attachment for information exchange. It ensures that information is exchanged in an orderly and understandable way where the information content of each informative dialogue act is connected with an entity introduced in the preceding context and, if there is no preceding context or if a new connected dialogue fragment has to be opened, it introduces a new point for connection.

In terms of the information packaging theory, topic management serves the linkage part of information packaging. And in terms of the file card metaphor, a topic introduction act instructs the listener to evoke a specific file-card in his knowledge store, or to construct one in case the listener has no previous knowledge about it. A topic shift act instructs the listener to open another file card. A topic continuation makes the listener continue the information update on the same file card. By pointing the loci of update, topic management acts structure the information update.

In each case, topic management aims at restricting the discussion on a certain entity, its directly associated entities, and the information that is requested, asserted, etc. about it in the conversation (Grosz (1981), Sidner (1983)). As a result, the topic serves as a "context" or a "framework" for information update

**4379

1	I:	Informatie Schiphol	Schiphol Information						
2	S:	Ja, goedemo...middag	Yes, good mo...afternoon						
		mevrouw	madam						
3		Kunt u mij misschien	Can you tell me						
		ook zeggen							
4		is het toestel uit	Has the plane from		T_1				
		Dubrovnik,	Dubrovnik,		↓				
5		de JU 222,	the JU 222,		T_1				
					↓				
6		die om twaalf uur	that was supposed		T_1	-	C_1		
		twintig op Schiphol	to arrive at Schiphol		↓				
		zou komen,	at twenty past twelve,		↓				
7		is die al geland?	has that yet landed?		T_1	-	C_2		
8	I:	Even kijken,	Let's see,						
9		een ogenblikje	just a moment						
10	S:	Alstublieft	Thank you						
11	I:	Hallo	Hello						
12	S:	Ja, mevrouw	Yes, madam						
13	I:	Nou, ik heb wel	Well, I have had	T_2 -	$C_3(\geq T_1)$				
		de JU 222 gehad,	the JU 222,		↓				
14	S:	Ja,	Yes,		↓				
15	I:	maar die komt niet	but it doesn't come		T_1	-	C_4		
		vanuit Dubrovnik	from Dubrovnik		↓				
16	S:	O,	Oh,		↓				
17		waar kwam die dan	where did it then		T_1	-	C_5		
					↓				
18		uit Zagreb?	from Zagreb?		T_1	-	C_6		
19	I:	Ja	Yes		↓		↓		
20	S:	Ja, das ook goed	Yes, that's all right too	$T_3(=[$	T_1	-	$C_6])$	-	C_7
					↓				
21	I:	Ja, die is geland hoor	Yes, it has landed		T_1	-	C_8		
					↓				
22		kwart voor een	a quarter to one		T_1	-	C_9		
					↓				
23	S:	Kwart voor een	A quarter to one		T_1	-	C_9		
24		Fijn,	Fine,						
25		dank u wel	thank you very much						
26	I:	Tot uw dienst hoor	You are welcome						
27	S:	Dag mevrouw	Goodbye madam						
28	I:	Dag mevrouw	Goodbye madam						

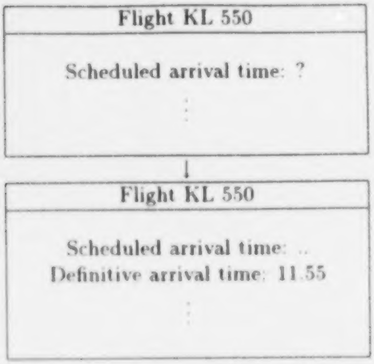

Flight KL 550
Scheduled arrival time: ?
Flight KL 550
Scheduled arrival time: ..
Definitive arrival time: 11.55

Figure 1: The information update in dialogue **2063

and as such enables the speakers to use informationally incomplete expressions in an unambiguous way.

We will illustrate this with the help of dialogue fragment **2063 and figure 1. Figure 1 describes the information update after each turn in the dialogue.

**2063

		...	
4		Vlucht KL 550,	Flight KL 550,
5		hoe laat is die gepland?	for what time is it scheduled?
6	I	Die wordt nu definitief verwacht om vijf voor twaalf	It is now definitely expected at five to twelve
7	S	Vijf voor twaalf?	Five to twelve ?
8	I	Ja hoor	Yes indeed
		...	

With utterance 4, the speaker introduces the topic of this short information exchange. In terms of the file metaphor, the speaker instructs the listener to open his file card of Flight KL 550. By doing this, he shows that he wants to restrict the exchange to this entity. In utterance 5, he asks the information that he wants to know about it, the scheduled arrival time. Since the framework of interpretation is already set, he refers to the topic with an anaphor, *it*.

The first box of figure 1 shows the update after the first turn. A file card of Flight KL 550 is opened and a slot on that card is highlighted. The speaker has made clear that he doesn't know the value of the slot.

With utterance 6, the information service gives the requested information. She even gives more information than asked for. Being very cooperative, she gives the information she considers more interesting for the information seeker (the definitive arrival time instead of the scheduled time). A pronoun is used to refer to the topic, since the framework of interpretation is clear. The second box of figure 1 shows the information update intended by this turn.

With utterances 7 and 8, the update is verified and grounded. The speakers abbreviate their utterances still more. In utterance 7, only the just updated in-

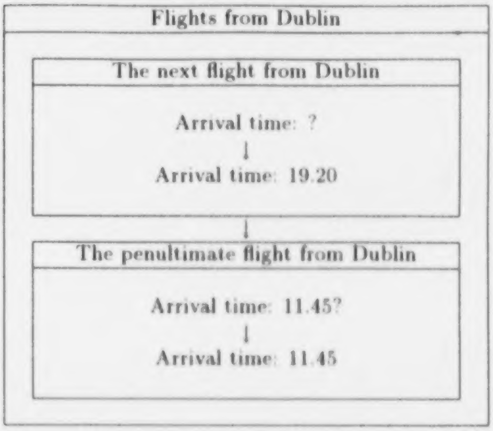


Figure 2: Topic change in dialogue **1144

formation element is expressed, which refers exactly to the information element that is checked. In utterance 8, only the most informative part of the answer is expressed *ja hoor (yes indeed)*. In both cases, the framework of interpretation, the topic, is presupposed.

Dialogue **1144 and figure 2 show an example of an information exchange in which a topic shift occurs.

**1144

		...	
5	S:	Zou je mij kunnen zeggen	Could you tell me
6		het eerstvolgende vliegtuig uit Dublin	the next plane from Dublin
7		wanneer dat aankomt?	when will it arrive?
8	I:	Dat is vanavond pas om twintig over zeven	That is this evening only at twenty past seven
9	S:	Negentien uur twintig	Nineteen hours twenty
10	I:	Ja	Yes
11	S:	Ja,	Yes
12		want eh.. het voorlaatste was zeker die een	because eh... I suppose the penultimate was that one
13		die om kwart voor twaalf aankwam	which arrived at a quarter to twelve
14	I:	Juist, ja ja	Right, yes yes
		...	

Utterance 6 of this dialogue introduces the first topic, *The first plane from Dublin.* It instructs the dialogue partner to open a file card of the first flight from Dublin. The description of the topic shows that it needs to be found within a bigger file-card named "Flights from Dublin".

With utterance 7, the arrival time of this flight is requested. It moves the attention of the listener to the arrival time slot on this specific file-card. Utterance 8 gives the value of the arrival time. It instructs the listener to update the arrival time slot with this value. Utterances 9 and 10 ground this instruction and make it mutually agreed[1]. The first embedded box of figure 2 represents this information update.

After this information exchange is closed, a new one is opened with the introduction of a new, although related topic, *the penultimate one*. This topic need also to be found within the scope of the more global topic "Flights from Dublin". Utterance 13 checks the arrival time of this topic, moving the attention of the dialogue partner to this particular slot on the file card. Utterance 14 confirms the check. The second embedded box of figure 2 represents the update process in this information exchange.

Both examples clearly show the function of topic management in an information dialogue. Topic management acts determine the locus of update. They restrict the attention of the speakers to this particular locus and as such enable the speakers to apply pronouns and ellipsis without ambiguity.

4 Integrating tail and focus

In Rats (1996), the analysis was restricted to topic management, and the information exchanged about topics was globally analysed as comment. With Vallduví's information packaging theory, the comment part can be refined and its function within the information exchange can be made more precise. Following Vallduví, the notions of focus and a tail may be defined as follows.

> *An information unit, F, is the focus of a dialogue act, D, iff F is the information that actually has to be updated with respect to the topic.*
>
> *An information unit, L, is the tail of a dialogue act, D, iff L refers to a characteristic of a topic the value of which need to be added, revised, checked etcetera.*

Of course, not all comments will contain of a tail. But all of them will contain a focus.

Which specific update a focus causes within the information update, must be derived from the communicative function that the utterance expresses. The focus of a wh-question is the information that is asked about the topic, as figures 1 and 2 illustrate (compare Hoepelman, Machate, and Schnitzer (1991)). In file card metaphor terms: the focus refers to a slot, the speaker doesn't know the value of. In principle, the focus of a wh-answer will be the item that gives the value of the slot[2](Hoepelman, Machate, and Schnitzer (1991)).

A wh-answer is a more specific variant of an inform, a dialogue act that intends to give information that is considered to be new for the listener. The focus of an inform is the specific information that the speaker considers to be new for the listener. It depends on the scope of the focus what update should take place. In case of wide focus, a slot needs to be created before it can be filled. In case of a narrow focus, a file-card and eventually a slot are considered to be available and the listener is instructed to update the slot with the value given by the focus of the inform.

---

[1] See Traum and Allen (1992) for a more extended explanation of grounding in dialogue.

[2] Of course, other reactions to a wh-question are possible. It could happen, for instance, that the other speaker doesn't know the value, so that a meta-dialogue will follow in which he explains that he doesn't know the answer. However, these kinds of reactions will not be defined as wh-answers.

As is argued by Hoepelman, Machate, and Schnitzer (1991), the focus of a yes/no-question or a check will be the item that the speaker asks the listener to verify for the topic. Hoepelman et al. give the following example dialogues to make their point[3].

(1) A: Is Dali a COMPOSER?
B: No,
he is a PAINTER

(2) A: Is DALI a composer?
B: No,
BEETHOVEN is a composer

In the first utterance of the first dialogue, the focus is *"COMPOSER"*. The dialogue partner is requested to verify if the characteristic *composer* holds for *Dali*. An important argument for this analysis is that if the question contains information that needs to be corrected, an utterance like *"he is a PAINTER"*, with *PAINTER* as the focus, expresses the felicitous correction. With this linguistic form, an alternative characteristic is given for Dali. The focus of the first utterance of the second dialogue is *DALI*. The dialogue partner is requested to verify if Dali belongs to the set of composers. The felicitous correction for this utterance is an utterance that gives an alternative member of this set. In both examples, the topic is kept the same during the update, while the foci form the dynamic part of the information exchange.

The focus of an alternatives-question is the list of alternatives that need to be checked for the topic, the focus of an alternatives-answer is one of the alternatives, and the focus of a correction is the item that needs to be corrected with respect to the topic. This is illustrated by example dialogue **5479 and figure 3.

**5479

6	S:	voor een eh intercontinentale vlucht	for an uh intercontinental flight	topic introduction
7		moet ik daar een uur of twee uur van te voren aanwezig zijn?	do I have to be present one or two hours in advance?	alternatives question
8	I:	Twee uur van te voren	Two hours in advance	alternatives answer
9	S:	Een uur van te voren?	One hour in advance?	check
10	I:	Nee,	No,	disconfirm
11		twee uur	two hours	correction

---

[3] The sentence elements in capital letters must be read as accented.

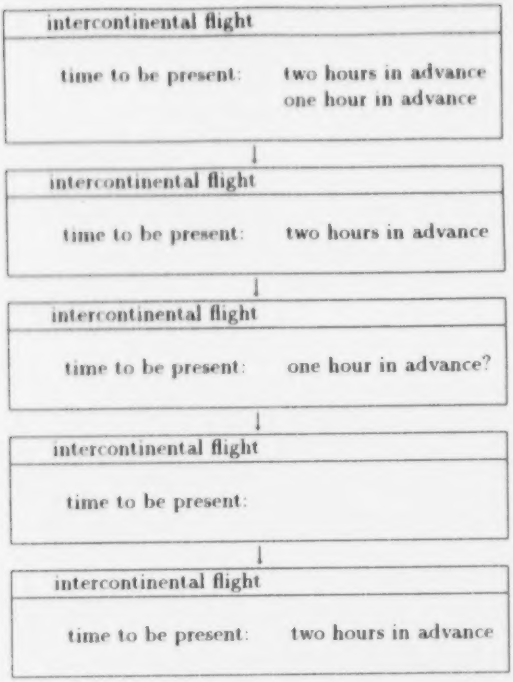


Figure 3: Information update in dialogue **5479

Utterance 6 of this dialogue introduces the topic of the information exchange, *"an intercontinental flight"*. Utterance 7 asks an alternatives question about it: if the speaker has to be present one or two hours in advance. The focus of this question, the items that need to be checked for the intercontinental flight, is *one hour in advance* and *two hours in advance*. The first box of figure 3 shows the update after this first turn. We see that the focus is represented as the possible values of a slot.

Utterance 8 gives the answer to the question. It only expresses the focus, since the context is given by the preceding turn. The second box of figure 3 shows the update aimed by the second turn. The value of the slot is changed in one of the alternatives.

Utterance 9 is a check. It shows how the speaker has understood the previous utterance. It only expresses the focus, since the framework of interpretation is still given. Box 3 in figure 3 shows which update is checked by utterance 9[4]. The other speaker doesn't agree with the value that is presented in the check, so with utterance 10 he performs a disconfirm. Box 4 gives the update aimed by utterance 10. The wrong value is taken from the representation. Utterance 11 gives the correct value. Again, only the focus is expressed to fill the empty slot. Box 5 of figure 3 shows the aimed update after this utterance.

---

[4] In fact, this particular step in the dialogue shows that a model for information exchange in dialogue should be slightly more complicated. We need to distinguish a representation of the information update of each of the speakers from the representation of the information update that is mutually agreed and understood (see for instance (Hoepelman, Machate, and Schnitzer

The examples show how the notion of comment can be refined by means of a focus and a tail. The topic and the tail form the framework for the information update, while the focus is the information element that is changing with each step in the information exchange.

5 Further research and conclusion

Now the theoretical framework is set, a study can be made to the linguistic realisation of topic, tail, and focus in naturally occurring conversations. The study will not only give us a better insight in the information packaging devices of the Dutch language, it will also enable us to find more empirical evidence for our framework. The first promising steps in this direction have been made.

In Rats (1996), an extensive study is reported of the syntactic realization of topic management in a corpus of 111 Dutch telephone conversations. It turns out that speakers apply special syntactic structures to mark changes in the topical structure of their conversation and follow standard word order in case of topic continuation. Rats and Bunt (1997) describes a study of the syntactic realization of focus in the same corpus. Also in this case, Dutch speakers apply special syntactic structures to mark the focus of their utterance, although the means are not as rich as for topic management.

The description of the syntactic realization of focus shows that more research is required into its prosodic realization. We saw for instance that to be able to know exactly which item is checked in case of a yes/no question, we need information about the placement of the sentence accent. Also for a complete description of the linguistic realization of topic management, we may need prosodic information. Research done to the relation between accentuation of referential expressions and topic management in spoken monologues (Terken (1984), Nakatani (1995)) in English and Dutch has shown that speakers indicate topic introductions and topic shifts by accentuation. So, it is plausible that new insights may be gained in this field too.

From these results, we may conclude that the study of information packaging in naturally occurring dialogues is worth to be studied. It enables us to extend and refine our insights about information update in dialogue, and it gives us a framework of interpretation for speakers' use of special syntactic constructions, abbreviate expressions and certain intonation contours.

(1991))). This complication is, for practical reasons, kept out of the scope of this paper.

References

Bunt, H. (1994). Context and dialogue control. *Think 3*, 19–31.

Chafe, W. (1976). Givenness, contrastiveness, definiteness, subjects, topics, and point of view. In C. Li (Ed.), *Subject and Topic*, pp. 25–55. Academic Press.

Chafe, W. (1987). Cognitive constraints on information flow. In R. Toulmin (Ed.), *Coherence and Grounding in Discourse*, pp. 21–51. John Benjamins Publishing Company.

Grosz, B. (1981). Focusing and description in natural language dialogues. In A. Joshi, B. Webber, and I. Sag (Eds.), *Elements of Discourse Understanding*, pp. 84–105. Cambridge University Press.

Heim, I. (1983). File change semantics and the familiarity theory of definiteness. In R. Bäuerle, C. Schwarze, and A. von Stechow (Eds.), *Meaning, Use, and Interpretation of Language*, pp. 164–189. De Gruyter.

Hoepelman, J., J. Machate, and R. Schnitzer (1991). Intonational focusing and dialogue games. *Journal of Semantics 8*, 253–275.

Nakatani, C. (1995). Discourse structural constraints on accent in narrative. In J. Santen, van, R. Sproat, and J. Olive, J. and. Hirschberg (Eds.), *Current Approaches in Speech Synthesis*. Springer Verlag.

Rats, M. (1996). *Topic Management in Information Dialogues*. Ph. D. thesis, Tilburg University.

Rats, M. and H. Bunt (1997). Information packaging in dutch information dialogues. To appear in the proceedings of the HCM workshop Discourse and Spoken dialogue.

Sidner, C. (1983). Focussing in the comprehension of definite anaphora. In M. Brady and R. Berwick (Eds.), *Computational Models of Discourse*, pp. 267–330. MIT Press.

Terken, J. (1984). The distribution of pitch accents in instructions as a function of discourse structure. *Language and Speech 27*(3).

Traum, D. and J. Allen (1992). A "speech acts" approach to grounding in conversation. In *Proceedings International Conference on Spoken Language processing (ICSLP-92*, pp. 137–140.

Vallduví, E. (1990). *The Informational Component*. Ph. D. thesis, University of Pennsylvania.

Vallduví, E. and E. Engdahl (1994). Information packaging and grammar architecture. In *Proceedings of NELS*. University of Pennsylvania.

ANNO: a Multi-functional Flemish Text Corpus

Ineke Schuurman*

Abstract

In this paper the ANNO Project ("Een Geannoteerde Publieke Gegevensbank voor het Geschreven Nederlands/An Annotated Database for Written Dutch") is reported on[1]. The project aims at laying the foundations for the compilation and linguistic annotation of a large multi-functional Flemish text corpus. The corpus available now consists of language written to be spoken, together with transcribed interviews.
In this paper we present the levels of annotation ANNO comes with at the moment. In general, we will show what can be achieved using taggers, parsers etc. that are currently available for Dutch. A separate issue is whether the tools are as useful for Flemish as they are for Dutch.

Introduction

The ANNO Project is sponsored by the Flemish Research Initiative in Speech and Language Technology. It is a pilot project, aiming at laying the foundations for the compilation and linguistic annotation of a large, multi-functional, standard Flemish text corpus.
Although great efforts have been made in creating machine-readable corpora for English and other major languages, this is only to a lesser degree the case for Dutch. To some extent this is understandable: the market for English NLP products is much larger than that for Dutch NLP products. On the other hand, to safeguard the position of languages like Dutch, Danish etc. both inside the European Union and beyond it is important to develop tools for the automatic processing of these languages as well: taggers, parsers, speech interfaces, etc. Otherwise, these languages are in danger of being pushed aside in our digitized society. For such reasons national governments, the European Union and other bodies promote the development of tools and resources for minor languages as well. As annotated corpora provide an excellent basis for developing NLP tools, corpora of reasonable size should be created for languages like Dutch as well, cf. also Kruyt (1995).

---

*Centrum voor Computerlinguistiek, K.U. Leuven

[1] One way or another the following people were also involved in ANNO: Joyce de Booy, Frank van Eynde, Wim Peters and Bruno Tersago.

1 Two variants of standard Dutch

According to the constitution the official language in Flanders is Dutch, just as it is in the Netherlands. So why should there be a corpus of Flemish[2]? Is standard Belgian Dutch different from standard "Dutch" Dutch? Yes, assuming that the language used on radio and television reflects the standard language[3].
Although many speakers of Dutch and Flemish are unaware of this it turns out that there are differences at many levels: phonology, morphology, syntax, semantics, pragmatics).
Some examples:

- Voicing of syllable-initial fricatives
- Stress patterns
- Other past tenses (Flemish *zegden* – Dutch *zeiden* (**said**)) and plurals (Flemish *leraars* – Dutch *leraren* (**teachers**))
- Gender. In Flemish there are three genders (*masculine*, *feminine* and *neuter*), in Dutch only two genders are left (*neuter* and *non-neuter*)
- The behaviour of separable verbs. In Flemish the separable affix often remains with the verb also in cases where this would be 'ungrammatical' for speakers of Dutch, cf. Hoekstra (1987, 35):

  (1) Hij aanhoorde het vonnis onbewogen (Fl)
  (2) Hij hoorde het vonnis onbewogen aan (D and Fl)
  'He listened to the sentence without emotion'

- The occurrence of Verb Projection Raising in Flemish:

  (3) ..., omdat zij wil een appel eten (Fl)
  (4) ..., omdat zij een appel wil eten (D and Fl)
  'because she wants to eat an apple'

- The choice of the auxiliary of the perfect. For a range of verbs in Flemish the choice of the auxiliary of the perfect depends on the main verb:

  (5) Hij heeft haar komen afhalen (Fl)
  (6) Hij is haar komen afhalen (D and Fl)
  'He came to fetch her'

---

[2] In this paper the notion *Flemish* will be used to refer to standard Belgian Dutch.
[3] See also Hoekstra (1987)

There are also reasons to believe that the distribution of the present perfect and the imperfect past to express that something happened before the moment of speech is not the same in both variants of Dutch (temporal semantics), whereas the same holds for the choice of the personal pronoun *jullie* or *je* vs *u* (**you** pl and sg). And of course there are the differences with respect to the vocabulary.

A sufficiently large corpus of Flemish, especially when contrasted with the same kind of corpus for Dutch, will also tell us more about these and other particularities of the language used.

It will be clear that, although in general both variants have the same properties, there is a whole number of phenomena which are 'out' in one of the variants of Dutch whereas they are perfect in the other variant. Take the role of gender: in Flemish one should use the genders correctly, one should for example refer to a bus with *zij* as it is a feminine noun. In Dutch people will not be aware of its feminine genus, therefore it often will be referred to as *hij*.

Thus far corpus linguistics didn't pay much attention to the variant used in Belgium.

No corpus of reasonable size at all was available in machine-readable format. The only completely Flemish, i.e. standard Belgian Dutch, corpus we are aware of is the one collected by Willy Martin (Martin (1967), cf. also Dutilh-Ruitenberg (1992)).

2 The objective of the project

The objective of the ANNO Project was twofold:

- the inventory of corpora, taggers, parsers, etc. that are available, especially for Dutch and Flemish;
- the compilation of a multi-functional database for Flemish, containing a corpus with a series of annotation schemes representing various levels of linguistic analysis

With respect to the second task: at this moment texts are annotated for their part-of-speech, morphological, syntactic and phonological information, and discourse information.

The tools to be used are preferably freely available for research purposes and have a good performance: correction of output is very time-consuming.

Another initial requirement was platform independence, i.e. the ANNO database should be usable in both DOS and UNIX environments[4].

3 Inventory

Our inventory, cf. the first objective (reported on in Peters and Tersago (1996)), showed that there are quite a number of corpora for Dutch, and the same holds for tools to treat them. But, as we expected, there was almost nothing available for Flemish.

---

[4] During the project we learned about JAVA, therefore the new objective is to make ANNO available on the Web.

Peters and Tersago (1996) contains chapters (in Dutch) on the design and compilation of corpora, on annotations, existing corpora, tools and recent initiatives. Several of these are made available on the Web[5].

The outcome of the inventory also to a large extent determined the choice of our tools.

4 Corpus

4.1 Composition of the corpus

As is clear from the full project title "Een Geannoteerde Publieke Gegevensbank voor het Geschreven Nederlands", ANNO[6] is an annotated corpus for *written* Dutch. Still the texts it contains are transcriptions of radio news and current affairs broadcasts, i.e. *spoken* language[7].

More specifically, ANNO contains texts

- with a wide circulation,
- intended for a broad population,
- treating non-specialist topics, and
- as recent as possible (Kruyt and Putter (1992), Martin, Platteau, and Heymans (1985))

The text material the ANNO corpus consists of has been derived from BRTN (Belgian Radio and Television) radio news broadcasts and the current affairs programme Actueel[8]: language written to be spoken together with transcribed interviews. The latter contain spontaneous speech.

4.2 Some obstacles

The BRTN-texts are not available in electronic format, so we had to scan several thousands of sheets of paper as every item is written on a separate sheet. A very time-consuming job by which also a considerable amount of structural (scanning) errors is introduced. These were corrected in a semi-automatic way.
The texts we received were not meant to be made public: the texts contain many

---

[5] http://www.ccl.kuleuven.ac.be/about/ANNO/inleiding.html

[6] In what follows the notion ANNO is used to refer to the whole project as well as to the corpus and/or the resulting database.

[7] A database of spoken Flemish as such is taken care of by another project within the Flemish programme for speech- and language processing, FONILEX.

[8] News: 21 - 26 March 1995, 17 - 30 April 1995, 1 - 30 May 1995 and 12 - 30 June 1995 , always the 08.00, 13.00, 18.00 and 24.00 broadcasts; Actueel: 20 - 29 March 1995, 1 - 31 July 1995, 1 - 31 August 1995, the 13.00 and 18.00 broadcasts (no broadcasts on Sundays and on holidays). A quite similar corpus for Dutch is described in Sterkenburg (1989).

typing errors and the spelling is very inconsequent (both preferred and alternative spelling within one item, many inaccuracies, even the names of the reporters themselves are written in three, four ways). Whenever the spelling didn't influence pronunciation we normalized the texts (preferred spelling) in order to simplify consultation of the corpus by future users[9].

07mei13u: binenland ⟶ binnenland

However, sometimes a word was 'misspelled' deliberately as a pronunciation help for the newsreader: *biezonder*, *honderste* and *Andaloesisch* instead of *bijzonder*, *honderdste* and *Andalusisch*. Such 'mistakes' are preserved as the newsreaders apparently tried to avoid a spelling pronunciation of these words: their pronunciation had to sound natural.
Abbreviations are always spelled out, as they will be used in their full form in the broadcasts themselves:

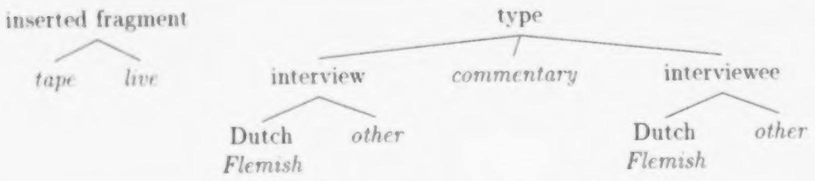
m/s ⟶ meter per seconde

One may question our approach with respect to these phenomena: we tried to come as close as possible to what was actually said (and how it was said), although we didn't have the tapes. Of course the original texts (without interventions from us) will be made available as well, whereas all our interventions (or lack of interventions, cf. the pronunciation help) will be motivated in the documentation that comes with the database. And all interventions are recorded in SGML-annotations.

The lack of tapes also complicated the encoding of the corpus in SGML as it was not always clear whether a paragraph belonged to an interview or was part of the text of the newsreader.

4.3 Typical properties of the texts involved

Typical for the Flemish news broadcasts as they are incorporated in our corpus is that they are read by two newsreaders and that they contain (live) interviews and commentaries. These inserted news fragments can be in a foreign language. Some of the inserted fragments are live, others are taped. Within both types *interviews* show an interaction between interviewer and interviewee, *commentaries* often contain prepared speech whereas speech fragments containing only statements by *interviewees* are often spontaneous.



---

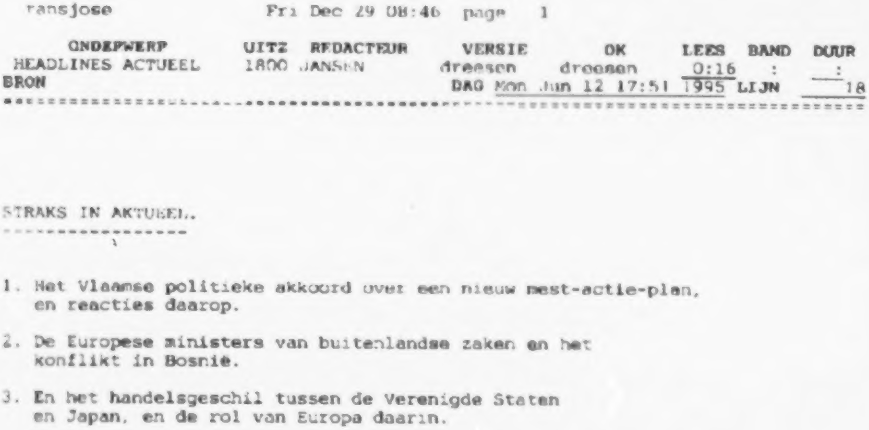
[9] When in doubt nothing was changed. Note that we couldn't consult the tapes as the BRTN did not want us to have them.

The same distinctions hold for Actueel, be it that the items are much longer and have a larger share of spontaneous speech.

Next to what you hear when listening to the radio, the corpus also contains a considerable amount of text-not-to-be-read-aloud: directions for the newsreader, administration, etc., see the following fragment (LW means "last words of the tape").

> 01mei08u: ...., maar als je wil kampioen worden dan moet je dat gewoon presteren, drie keer winnen.
>
> LW gewoon presteren, drie keer winnen.

Another example is the header in Figure 1 .



```
ransjose                 Fri Dec 29 08:46  page   1

     ONDERWERP           UITZ  REDACTEUR      VERSIE        OK        LEES  BAND  DUUR
 HEADLINES ACTUEEL       1800 JANSEN        dreesen    dreesen       0:16    :      :
BRON                                         DAG Mon Jun 12 17:51 1995 LIJN        18
=====================================================================================


STRAKS IN AKTUEEL.
-----------------

1. Het Vlaamse politieke akkoord over een nieuw mest-actie-plan,
   en reacties daarop.

2. De Europese ministers van buitenlandse zaken en het
   konflikt in Bosnië.

3. En het handelsgeschil tussen de Verenigde Staten
   en Japan, en de rol van Europa daarin.
```

Figure 1: Part of an original text: 12 June 1995, 18h: the headlines of Actueel

5 Annotation

In this section the various types of annotation will be discussed. Often tools require their input to be in a well-defined format (without accents, without ASCII-codes, etc.), each tool having its own desiderata. Several small AWK-programmes had to be written to convert the corpus into the desired formats.

5.1 Standard Generalized Markup Language

By means of SGML-codes all information in the corpus is captured unambiguously, cf. Sperberg-McQueen and Burnard (1994), Ide and Véronis (1994). When scan-

170

ning texts and/or transferring the corpus to another platform the lay-out of the texts may change. The SGML-codes will tell you exactly how the original texts looked like. In the following example part of the news broadcast of 21mei08u is reproduced without and with codes. In this case only *representative* information is involved:

> In de Burundese hoofdstad Bujumbura loopt de etnische spanning op. Bij nieuwe gevechten vannacht zijn er opnieuw doden gevallen.
>
> In Tokio zijn nu al acht doden geteld na de aanval met sarin-gas in de metro. Volgens een Japanse ochtendkrant zou één verdachte zijn geïdentificeerd; de politie gaat ervan uit dat er een georganiseerde bende aan het werk is geweest.
>
> De Franstalige socialisten willen dat premier Dehaene bemiddelt in het dispuut rond de uitbouw van communicatie-netwerken in ons land.

```
<div1 ID=210508.2><HEAD>Headlines<HEAD>
<p>
<list type=simple>

<item> In de Burundese hoofdstad Bujumbura loopt de etnische spanning
op. Bij nieuwe gevechten vannacht zijn er opnieuw doden gevallen. </item>

<item> In Tokio zijn nu al acht doden geteld na de aanval met sarin-gas in de
metro. Volgens een Japanse ochtendkrant zou &eacute;&eacute;n verdachte
zijn ge&iuml;dentificeerd ; de politie gaat ervan uit dat er een georganiseerde
bende aan het werk is geweest. </item>

<item> De Franstalige socialisten willen dat premier Dehaene bemiddelt
in het dispuut rond de uitbouw van communicatie-netwerken in ons land.
</item>
</list>
</p>
</div1>
```

Interpretative information is to be coded as well. In the following fragment the dots indicate that the newsreader has to wait a few moments before he completes the sentence (the listener is informed that this time Ireland didn't win the European Song Contest)(14mei13u):

> Of toch niet helemaal. Het winnende nummer, Nocturne van de groep Secret Garden, heeft maar een tekst van 24 Noorse woorden. De rest van het nummer is een vioolsolo, gespeeld door . . . een Ierse violiste.
>
> Of toch niet helemaal. Het winnende nummer, Nocturne van de groep Secret Garden, heeft maar een tekst van 24 Noorse woorden. De rest van het nummer is een vioolsolo, gespeeld door <pause>...</pause> een Ierse violiste.

A series of dots may also mean that the transcriber didn't understand what was said. In such cases a correct sentence was constructed for linguistic annotation as the original construction will have been correct:

> - Met Swissair hebben we meer bepaald beslist dat onze streefdoelen competitiviteit, kwaliteit en winst zullen zijn. ... zullen zo snel mogelijk en maximaal verwezenlijkt worden.

becomes

> - Met Swissair hebben we meer bepaald beslist dat onze streefdoelen competitiviteit, kwaliteit en winst zullen zijn. **Deze** zullen zo snel mogelijk en maximaal verwezenlijkt worden.

In the SGML-coded original the gap is respected:

```
<int> <speaker> – </speaker><p>Met Swissair hebben we meer be-
paald beslist dat onze streefdoelen competitiviteit, kwaliteit en winst zullen
zijn.<gap reason="inaudible" resp="transcriber"><completion>Deze</completion
zullen zo snel mogelijk en maximaal verwezenlijkt worden.< /p>< /int>
```

It will be clear that coding texts in SGML the way described above will always involve human interference. Our decisions in this matter may be questioned, especially with respect to our treatment of gaps. We have opted for this solution in order to give our tools a fair chance. The completions are always as neutral as possible. And of course the original texts are available as well.

As remarked before the whole corpus was tagged with SGML, including the parts in a foreign language. These parts, however, have been taken out of the corpus when it comes to linguistic annotations as we didn't have the means to treat these.

This means that of a fragment like the following only the first and the last paragraph are annotated for part of speech, phonology etc.

> De uitslag van de verkiezingen die vandaag beginnen zal bijzondere aandacht krijgen op de verschillende politieke hoofdkwartieren.
>
> Das Oberkommando der Wehrmacht gibt bekannt: Seit mitternacht schweigen nun an allen Fronten die Waffen auf Befehl des Grossadmirals ...
>
> I only wish that Franklin Lee Roosefelt[10] had lived to witness this day. General Eisenhower informs me that the forces of Germany have surrendered to the United Nations. The flags of freedom fly all over Europe.
>
> U hoorde eerst een Duitse omroeper, en daarna de Amerikaanse president Truman, die elk op hun manier het officiële einde afkondigden van de Tweede Wereldoorlog in Europa. Dat is vandaag precies vijftig jaar geleden.

With SGML-annotation this looks like:

---

[10] Cf. note 9 about misspellings.

<p>De uitslag van de verkiezingen die vandaag beginnen zal bijzondere aandacht krijgen op de verschillende politieke hoofdkwartieren.< /p>
<int><lang=german><p>
Das Oberkommando der Wehrmacht gibt bekannt: Seit mitternacht schweigen nun an allen Fronten die Waffen auf Befehl des Grossadmirals <gap reason="inaudible" resp="transcriber">< /p>
< /lang><lang=english><p>
I only wish that Franklin Lee Roosefelt had lived to witness this day. General Eisenhower informs me that the forces of Germany have surrendered to the United Nations. The flags of freedom fly all over Europe.< /p>< /lang>< /int>
<p> U hoorde eerst een Duitse omroeper, en daarna de Amerikaanse president Truman, die elk op hun manier het offici&euml;le einde afkondigden van de Tweede Wereldoorlog in Europa. Dat is vandaag precies vijftig jaar geleden.< /p>

5.2 Part-of-speech annotation

WOTAN (WOordklasse TAgger voor het Nederlands), cf. Berghmans (1994), is a POS-tagger developed at the University of Nijmegen on basis of the TOSCA-tagger for English. The tagset is based on Geerts, Haeseryn, de Rooij, and van den Toorn (1984) and satisfies the EAGLES-standard[11] for corpus annotation, also with respect to their *recommended* tagset. Next to its quite reasonable performance for Dutch, these features made WOTAN an attractive candidate for us.

The tagset of WOTAN distinguishes 10 main word classes (plus 2 additional ones):(**N**oun, **V**erb, **Art**icle, **Adj**ective, **Adv**erb, **Num**eral, **Prep**osition, **Pron**omen, **Conj**unction, and **Int**erjection (plus **Punc**tuation and **Misc**ellaneous). They all come with further specifications (person, number, gender, valency, case, etc.). One of these further specifications concerns the way the element is used: attributive, substantive, or adverbial. As many mistakes are due to this distinction, the developers of WOTAN suggest to leave this feature out in future. As this distinction is not recommended by EAGLES either, it is not included in the reduced WOTAN tagset with which the complete corpus is tagged (see also Schuurman and Tersago (1996), and the ANNO webpages). An example with both tagsets:

In de Burundese hoofdstad Bujumbura loopt de etnische spanning op.
(21mrt08u.txt, sentence 6)

---

[11] EAGLES: Expert Advisory Group on Language Engineering Standards. EAGLES is part of the LRE programme of the EU (DG-XIII). The EAGLES recommendations are to be found at http://www.ilc.pi.cnr.it/EAGLES96/browse.html

	full tagset	reduced tagset
∧		
In	Prep(voor)	Prep(voor)
de	Art(bep,zijd_of_mv,neut)	Art(bep,zijd_of_mv,neut)
Burundese	Adj(attr,stell,verv_neut)	Adj(stell,verv_neut)
hoofdstad	N(soort,ev,neut)	N(soort,ev,neut)
Bujumbura	N(eigen,ev,neut)	N(eigen,ev,neut)
loopt	V(intrans,ott,3,ev)	V(intrans,ott,3,ev)
de	Art(bep,zijd_of_mv,neut)	Art(bep,zijd_of_mv,neut)
etnische	Adj(attr,stell,verv_neut)	Adj(stell,verv_neut)
spanning	N(soort,ev,neut)	N(soort,ev,neut)
op	Adv(deel_v)	Prep(op)
.	Punc(punt)	Punc(punt)

Note that in the reduced version of WOTAN the separable verbal particle **op** is considered to be a preposition, a simplification suggested by the developers because too many mistakes were made. This is to be corrected by hand if so desired. Within the ANNO project this was corrected indeed.

In both tagsets WOTAN makes use of so-called **portmanteau** tags like **zijd_of_mv** (non-neuter or plural) or **hulp_of_kopp** (auxiliary or copula).

For Dutch the performance when using the full tagset is claimed to be 90 % at the level of the tags, and 95 % at the level of the word class for the extended tagset (for the reduced tagset the performance comes close to 94 % for the tags). Post-editing is therefore necessary.
The scores (full tagset) for our Flemish corpus were not that good: 86 % at the level of the tags and 94 % at the level of the word class[12]. Analysis of the mistakes showed us that many mistakes are made in constructions with typical Flemish properties (order of verbs, verb projection raising, colloquial speech). Ideally the tagger should be adapted to Flemish.

5.3 Phonological annotation

The complete corpus comes with phonological annotations by means of TreeTalk (beta version), a grapheme-to-phoneme conversion tool developed at the Universities of Antwerp and Tilburg.
Its output is in YAPA (Yet Another Phonetic Alphabet) which is IPA in 7-bits ASCII. It is developed at the K.U.Leuven and will be used by all projects within the programme "Spraak- en Taaltechnologie". It is to reflect the Flemish pronunciation.
The conversions by TreeTalk are not corrected. At the moment the idea is just to give the user an indication of the kind of phonological annotation we have in mind for the future. TreeTalk is first to be improved (for example on basis of the outcome of the aforementioned FONILEX project). Correction by hand was infeasible within the current project.
As far as we are aware TreeTalk is the only tool available to get phonological

---

[12] Note that one can not just compare the scores as the composition of the corpora involved is different. The WOTAN corpus consists of newspapers.

annotation for Flemish (The CELEX database, for example, reflects the Dutch pronunciation! And especially for phonological annotation one can not work with tools for Dutch-in-general. The relation between grapheme and phoneme in both language variants is not the same.)

> De Verenigde Naties zijn er niet in geslaagd om in Bosnië het bestand te verlengen dat vanmiddag afloopt.

de	d@	bosnie	bOsniE
verenigde	v@ren@Gd@	het	@t
naties	nasis	bestand	b@stAnt
zijn	zE"	te	t@
er	@r	verlengen	v@rlEN@n
niet	nit	dat	dAt
in	In	vanmiddag	vAnmIdAx
geslaagd	G@slaxt	afloopt	Aflopt
om	Om	.	@
in	In	^	@

5.4 Morphological annotation

It was quite difficult to find a morphological tagger for Dutch. Asking around on the net resulted in two candidates XSoft (Xerox) and KEPER (Polderland). XSoft turned out not yet to be available at the moment we needed it, therefore we only considered KEPER. It soon turned out that its functionality was not what we were looking for. We just needed in three fields 1) the item itself, 2) the lemma and 3) its internal structure (with special features, cf. below).

Therefore it became rather unappealing to tag the whole corpus with KEPER. Instead we developed our own tagset (AnnoMorf), which was applied to a very small part of the corpus (as tagging by hand is very time-consuming). This exercise gave us the possibility to adjust the tagset. AnnoMorf makes use of both the CELEX-database and the outcome of WOTAN.

In the third field for verbs not the 'neutral' stem should be given (that is already contained in the second field) but the past stem (like *zou*) or the participle stem (like *bombardeer*), TENSE meaning present tense affix, PTENSE past tense affix, PASTP past participle affix, etc. (cf. Schuurman (1997)):

zouden\zal\zou+PTENSE\
kunnen\kan\kan+TENSE\
gebombardeerd\bombardeer\bombardeer+PASTP\
gestegen\stijg\steeg+PASTP\

A tool with this functionality is under construction. In a later version another functionality should be added as well: of complex words it should be made clear what is the status of the boundary when no connective sound (as in "voorjaarS-buien") is involved:

voorjaarsbuien\voorjaarsbui\voorjaar+S+bui+EN\
aardbeving\aardbeving\aarde+beving
regelgeving\regelgeving\regel+geef+ing
media\medium\medium+PL\

Note that in "regelgeving" (issuing of rules) the part "geving" is not a word in Dutch, whereas in "aardbeving" (earthquake) both "aarde" and "beving" are existing words. In "voorjaar" (spring) both parts do exist as separate words, but still the word "voorjaar" is to be considered a simplex word.

5.5 Syntactic annotation

The syntactic annotation should add two further clues:

- constituents
- functions fulfilled by the constituents

In ANNO part of the METAL-parser developed by Siemens-Nixdorf was used in order to obtain a flat, bracketed structure (cf. the recommendations by EAGLES, section 1.3.3.2
(URL: http://www.ilc.pi.cnr.it/EAGLES96/browse.html.)), enriched with syntactic functions like *Subject*, *SCOMP*, etc[13]. METAL was chosen because it is the only syntactic parser for Dutch we are aware of yielding a flat, bracketed structure. As the results were not what we expected them to be[14] we will move over to another syntactic parser, probably one based on AGFL[15] or on ALEP[16]. In parallel a tool taking care of so-called *partial parsing* should be taken care of.

Below an example parsed with METAL: 21mrt08, sentences 2 and 6. Note that in sentence 2 some words (16/19) are not included in any constituent, nor are they considered constituents themselves. METAL is robust enough not to fail when it cannot handle part of the input. On the other hand there were too many sentences not receiving any constituent structure at all. Of course, everything can be corrected by hand. But as soon as there are too many 'mistakes' this is not feasible from a practical point of view.

> Het KMI verwacht vooral in het westen van het land mooie opklaringen, elders af en toe ook bewolking.
>
> In de Burundese hoofdstad Bujumbura loopt de etnische spanning op.

---

[13] At the moment METAL is distributed by LANT and it is called LanTmark.

[14] In fact we made an improper use of the METAL technology: the rules in the METAL parser were written with other applications and other types of sentences in mind. It turned out not to be possible to adapt the parser to our needs, at least not during the project. This appears to be one of the drawbacks of working with a commercial product.

[15] "Affix grammars over a Finite Lattice" (AGFL) is developed in Nijmegen, at the Department of Software Engineering. For more information, cf. http://www.cs.kun.nl/agfl/

[16] The "Advanced Language Engineering Platform" (ALEP) is an initiative of the European Commission. For more information, cf. http://www.iai.uni-sb.de/alep/

One problem concerns verbs with separable affixes as in "oplopen" (increase). In sentence 6 the affix is left out, other times it is considered a preposition used in postposition. Discontinuous structures in general present problems for the parser.

```
(2   [CLS [CLS [NP $SUBJ ("Het" 1) ("KMI" 2) ] [PRED
     ("verwacht" 3) ]
     [PP ("vooral" 4) ("in" 5) ("het" 6) ("westen" 7) ] [PP $POBJ
     ("van" 8) ("het" 9) ("land" 10) ]
     [NP $DOBJ ("mooie" 11) ("opklaringen" 12) ] ] ("," 13) [CLS
     ("elders" 14)
     [PRED ("is" 15) ] ("er" 16) ("af" 17) ("en" 18) ("toe" 19) [NP
     $SUBJ ("ook" 20) ("bewolking" 21) ]
     [PP ("met" 23) ("vooral" 24) ("in" 25) ("de" 26) ("Ardennen"
     27)
     [PP ("op" 30) ("nog" 28) ("kans" 29) ] ("lichte" 31)
     ("voorjaarsbuien" 32) ] ] ] ("." 33) )
(6   [CLS [PP $MOV ("In" 1) ("de" 2) ("Burundese" 3) ("hoofdstad"
     4) ("Bujumbura" 5) ]
     [PRED ("loopt" 6) ] [NP $SUBJ ("de" 7) ("etnische" 8)
     ("spanning" 9) ] ] ("." 11) )
```

5.6 Discourse annotation

In a last annotation round semantic information concerning Tense and Aspect is added. At the moment this is done by hand. Within the NFWO-project LINGUADUCT this approach will be worked out and implemented in ALEP.
Per sentence six types of information are given in just as many fields, cf. Booij (1996).

Field 1: TEMPORAL ANAPHORA
Does the *point of reference* of the sentence under consideration coincide with the point of reference in the previous sentence? **g** says that both points of reference are *simultaneous*, **n** that they are *not simultaneous*.
Field 2: TENSE
What is the relation between the *point of reference R* and the *point of perspective P*? **v** describes the relation as being *anterior*, **g** as *simultaneous* and **n** as *posterior*.
Field 3: TEMPORAL ADJUNCTS
In case the sentence contains a temporal adjunct this adjunct is qualified as being **l** (*locational*) or **r** (*relational*). If it is relational there is a further distinction in *deictic* (**d**) and *anaphoric* (**a**) ones. A third value tells whether the adjunct expresses *anteriority* (**v**), *simultaneity* (**g**) and *posteriority* (**n**) or whether it is to be considered a *general adjunct* (**a**).
Field 4: ASPECT
What is the relation between the *time of event E* and the *point of reference R*? **p** says it is *perfective*, **d** *durative*, **r** *retrospective*, **t** *terminative*, **i** *inchoative* and **pr** *prospective*.
Field 5: ASPECTUAL ADJUNCTS

Are the aspectual adjuncts to be classified as *durative adjuncts* (**d**) or as *frame adjuncts* (**g**)? Durative adjuncts are subdivided in *in*-adjuncts (**i**) and *for*-adjuncts (**f**), frame adverbials in adjuncts marking the *beginning* (**b**) or the *end* (**e**).

Field 6: AKTIONSART
Is the basic proposition *bounded* (**b**) or *unbounded* (**o**)?

For a sentence containing several finite clauses the information is expressed for all of these clauses. In such a case the values for the clauses is separated by a "+" (as shown in the second example). Note that in the fields 3 and 5 the values will be complex ones. On the other hand, they may remain empty since adjuncts are optional.

> In de Burundese hoofdstad Bujumbura loopt de etnische spanning op.
>
> \ n \ g \ \ d \ \ o
>
> Morgen blijft het nog aan de frisse kant, vanaf donderdag wordt het overdag heel wat zachter.
>
> \ n"+"n \ n"+"n \ rdn \ d"+"t \ gb \ o"+"o

5.7 Some figures

The full corpus, i.e. the corpus as it was scanned, contains approximately 646.500 words (± 4.2 MB), of which 340.000 words (2.2 MB) news broadcasts and 306.500 words (2 MB) Actueel.

The whole corpus is corrected for errors which may result from scanning. Of these 4.2 MB 2.65 MB is edited as described in section 4.2 (1.85 MB news, 0.8 MB Actueel).

SGML-codes have been added for all corrected texts, i.e. 2.65 MB.

Everything (± 4.1 MB as foreign text fragments were excluded) was tagged for part-of-speech with the reduced WOTAN-tagset, 2.65 MB was also tagged with the extended tagset (cf. section 5.2). Of this 2.65 MB 1.3 MB has already been corrected by hand.

0.5 MB is annotated for syntactic information with METAL (section 5.5) and 0.2 MB for morphological information. The latter was done by hand, cf. section 5.4.

The whole corpus is provided with a phonetic annotation (cf. section 5.3), the outcome is not corrected.

A small part of the corpus (0.07 MB) is also annotated for discourse information, more specifically for temporal information (Tense and Aspect). This was done by hand.

6 Conclusion

Creating a multi-functional, annotated linguistic database from scratch is quite a job. There is still a long way to go: tools should be adapted for Flemish (WOTAN),

others should be improved (TreeTalk) and further developed (AnnoMorf, the discourse tool). The whole corpus is to be parsed once more with another parser. We have the feeling that this duplication of work does pay off when we find a parser giving a better result. In that case the correction phase will be far less time-consuming. Remember that such a correction phase will return time after time! So it is worth the effort.
More text genres are to be added as well. At the moment we are collecting a subcorpus with texts from Flemish newspapers.

It will be clear that especially for phonological annotation one cannot work with tools for Dutch-in-general, we didn't even give such a tool a try. The relation between grapheme and phoneme is different in both language variants. Phonological information out of the CELEX database can not be used.
For other annotation tools the situation is less clear: the from our point of view unsatisfying performance of both METAL and KEPER is not to be attributed to the fact that Flemish texts were involved. They just don't satisfy our needs. On the other hand we have the impression, based on an error analysis, that the performance of WOTAN will be better when it is tuned for Flemish.

A last task will be to make everything available via the Web, making use of JAVA and *Abundantia Verborum* (see Speelman (1997)). A complication, however, is that the BRTN doesn't allow us to distribute their texts freely, at least not for commercial purposes. We will have to find a means to make as much as possible of the corpus public.

References

Berghmans, J. (1994). Wotan, een automatische grammatikale tagger voor het Nederlands. Master's thesis, Katholieke Universiteit Nijmegen.

Booij, J. d. (1996). Tense en Aspect in het Nederlands. Master's thesis.

Dutilh-Ruitenberg, W. (1992). Corpus Annotation Schemes in the Netherlands. INL Working Papers 92-03.

Geerts, G., W. Haeseryn, J. de Rooij, and M. van den Toorn (Eds.) (1984). *Algemene Nederlandse Spraakkunst.* Groningen/Leuven: Wolters-Noordhoff.

Hoekstra, E. (1987). Verb Raising and Verb Projection Raising in Flemish and Dutch. A report prepared for the Ministerie van de Vlaamse Gemeenschap.

Ide, N. and J. Véronis (1994). Corpus Encoding. Eagles Document EAG-CSG/IR-T2.1, EAGLES.

Kruyt, J. (1995). Nationale tekstcorpora in internationaal perspectief. *Forum der Letteren 36*(1), 47–58.

Kruyt, J. and E. Putter (1992). Corpus Design Criteria. INL Working Papers 92-11.

Martin, W. (1967). *De inhoud van krant en roman. Een frequentieonderzoek.* Antwerpen: Plantyn.

Martin, W., F. Platteau, and R. Heymans (1985). Naar een corpus voor een woordenboek hedendaags Nederlands. Mogelijkheden en beperkingen van het gebruik van corpora in lexicografisch onderzoek. UIA.

Peters, W. and B. Tersago (1996). Tekstcorpora. de stand van zaken. ANNO-project, Centrum voor Computerlinguïstiek, K.U.Leuven.

Schuurman, I. (1997). AnnoMorf. Centrum voor Computerlinguïstiek, K.U.Leuven.

Schuurman, I. and B. Tersago (1996). ANNO - IWT 940048. Wetenschappelijk Verslag, Centrum voor Computerlinguïstiek, K.U.Leuven.

Speelman, D. (1997). *Abundantia Verborum. A Tool for representing and presenting data of lexicological and lexicographic studies.* Ph. D. thesis, Katholieke Universiteit Leuven.

Sperberg-McQueen, C. and L. Burnard (1994). *Guidelines for Electronic Text Encoding and Interchange* (TEI P3 ed.). Chicago, Oxford: Text Encoding Initiative.

Sterkenburg, P. v. (1989). *Taal van het Journaal. Een momentopname van hedendaags Nederlands.* 's-Gravenhage: SDU-Uitgeverij.

GoalGetter: Predicting Contrastive Accent in Data-to-Speech Generation

Mariët Theune[*]

Abstract

This paper addresses the problem of predicting contrastive accent in spoken language generation. The common strategy of accenting 'new' and deaccenting 'old' information is not sufficient to achieve correct accentuation; generation of contrastive accent is required as well. I will discuss a few approaches to the prediction of contrastive accent, and propose a practical solution which avoids the problems these approaches are faced with. These issues are discussed in the context of GoalGetter, a data-to-speech system which generates spoken reports of football matches on the basis of tabular information.

Introduction

In language generation systems which produce spoken output, it is important to produce a natural sounding accentuation pattern for each generated sentence. Unnatural sounding speech output is unpleasant to listen to and may be difficult to understand. However, the accentuation pattern should not only be natural sounding but it should also be appropriate with respect to the meaning of the sentence.

In spoken language, accent placement has a major influence on interpretation. Sentences having the same surface structure but a different accentuation pattern may express very different meanings. A well-known example is the sentence *Mary only introduced Bill to Sue* (Rooth (1992)), which can have, among others, the following two accentuation patterns (accented words are given in italics)[1]:

(1) a Mary only introduced *Bill* to Sue
b Mary only introduced Bill to *Sue*

The accentuation patterns presented above each give rise to a different interpretation of the sentence. The accentuation pattern in (1)a indicates that Mary introduced only one person to Sue, and that person was Bill, whereas (1)b conveys that Mary introduced Bill to only one person, and that was Sue.

---

[*] This research was carried out at IPO, Center for Research on User-System Interaction, within the framework of the Priority Programme Language and Speech Technology (TST). The TST-programme is sponsored by NWO (the Netherlands Organization for Scientific Research).

[1] For the sake of clarity, in this and the following examples only relevant words are marked for accentuation; e.g., in (1)a/b it is irrelevant whether *Mary* is accented or not and therefore no accentuation is indicated for this word.

When automatically generating spoken output, it is essential that the accentuation pattern assigned to each sentence is in accordance with the intended meaning. If the example sentence *Mary only introduced Bill to Sue* should be interpreted as **∃!x[introduce(mary,x,sue)] & introduce(mary,bill,sue)**, for instance serving as an answer to the question *Who did Mary introduce to Sue?* pronouncing it as in (1)b would be inappropriate and cause the hearer to be confused. The hearer would be faced with conflicting information: the context of the utterance (= the preceding question) would suggest the interpretation given above, whereas the accentuation pattern would give rise to the interpretation **∃!x[introduce(mary,bill,x)] & introduce(mary,bill,sue)**.

It will be clear that, ideally, a spoken language generation system should always assign a correct accentuation pattern to the sentences it generates. In many cases, more than one accentuation pattern can be said to be 'correct', i.e., be in accordance with the intended meaning. What counts as a correct accentuation pattern depends on many factors, including the syntactic and semantic features of the output sentence and its relation to the discourse context. In this paper, I will concentrate on *contrast* as an important discourse semantic factor that must be taken into account for the generation of correct accentuation patterns. I will propose a way of detecting the presence of contrastive information and using this as a basis for the assignment of pitch accent. This will be done within the framework of the GoalGetter system, a data-to-speech system[2] which generates football reports from tabular data.

This paper is structured as follows. After a short introduction to GoalGetter, I will explain the system's original accentuation strategy and explain why this strategy sometimes produced incorrect accentuation patterns (section 1). Since I argue that this could be improved by adding contrastive accent, I will then discuss some existing approaches to contrast and show that these approaches are not attractive as a basis for implementation (section 2). After that, I discuss a practical method for the prediction of contrastive accent which has by now been implemented in GoalGetter, and could be implemented in other data-to-speech systems as well (section 3). In section 4, I discuss some future work. Finally, some conclusions are presented.

1 Accentuation in GoalGetter

Since GoalGetter is described in Klabbers (1997) (this volume), I will only give a very short overview of the system. For further details I refer to Klabbers et al. (1996), Klabbers et al. (1997) and Theune et al. (1997).

The GoalGetter system produces football reports in the form of a spoken monologue in Dutch. These reports are automatically generated on the basis of Teletext pages which contain tabular information on football matches played in the Dutch First Division. The system has two main modules, a language generation module (LGM) and a speech generation module (SGM). The LGM uses the football data from the input Teletext page to generate a written football report, which is annotated with prosodic markers, including accentuation markers. This annotated text is input to the SGM, which turns it into a speech signal. Since the assignment of accentuation markers is done in the LGM, I will give a brief description of this module only, restricting the description to those aspects which are relevant for accentuation.

---

[2] Such systems are sometimes called 'concept-to-speech' systems.

The input for the LGM is a table containing data on a particular football match, which are automatically derived from the information on a Teletext page. This table is converted into an internal data structure which has the form of a record with fields, as shown (partially) in Figure 1.

$$\left[\begin{array}{ll}
\text{teams:} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{teampair}} \\ \text{home\_team:} & \textit{teamtype} \\ \text{visitors:} & \textit{teamtype} \end{array}\right] \\
\text{goallist:} & \left\{\begin{array}{ll} \multicolumn{2}{l}{\textit{list\_of\_goalevents}} \\ \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{goal\_event}} \\ \text{team:} & \textit{teamtype} \\ \text{player:} & \textit{playertype} \\ \text{minute:} & \textit{integer} \\ \text{type:} & \textit{goaltype} \end{array}\right] & , \ldots \end{array}\right\} \\
\text{result:} & \textit{resulttype} \\
\text{cardlist:} & \textit{list\_of\_cardevents} \\
\text{referee:} & \textit{refereetype} \\
\text{number\_of\_spectators:} & \textit{integer}
\end{array}\right]$$

Figure 1: Data structure containing match data

The fields of this record can be expressed by one or more syntactic templates, which are syntactic tree structures containing slots for variable expressions. The filling of the slots depends mainly on conditions on the *Discourse Model*, which contains information about which linguistic expressions have been used in the preceding text, and what they referred to. Rules formulated in terms of this Discourse Model make it possible to use various referential expressions (proper names, pronouns, definite descriptions, etc.) appropriately. When a new sentence has been generated, the Discourse Model is updated accordingly.

The accentuation pattern of each generated sentence is determined on the basis of its syntactic structure and its relation to the preceding text. The accentuation algorithm is based on a version of Focus-Accent Theory (Dirksen (1992), Dirksen and Quené (1993)) and works as follows. First the system determines which parts of the generated sentence are out of focus and should therefore not be accented. This is done on the basis of information in the Discourse Model. Then, partly language-specific accentuation rules determine the distribution of accents, taking both the syntactic structure of the sentence and the focus information into account. Information about these syntax-based rules can be found in Theune et al. (1997). Here I will only discuss the semantic factors which are currently used to determine which phrases are out of focus.

In GoalGetter, a word or phrase will be regarded as being out of focus, and therefore not to be accented, for two reasons: if it is 'unaccentable' or if it conveys 'given' information. To determine if a word is unaccentable, the system simply checks if it belongs to a pre-defined list of words which normally do not receive an accent, e.g., certain function words. The second case is more interesting. As was observed by Halliday (1967), Chafe (1976), Brown (1983) and others, accent can function as a marker of information status: phrases expressing 'new' information are normally accented, while phrases expressing 'given' (or 'old') information are not.

In order to exploit this relationship between accent and information status, the GoalGetter system uses rules to determine whether a certain phrase expresses given information. These rules are based on the theory proposed by van Deemter (1994), who distinguishes two kinds of givenness: object-givenness and concept-givenness. A phrase is regarded as object-given if it refers to a discourse entity that has been referred to earlier in its local discourse domain, which in the present implementation consists of all preceding sentences in the same paragraph. Whether this situation holds can be checked in the Discourse Model. The following fragment can serve as an illustration.[3]

(2) a In the fifth minute, Kluivert scored a goal for Ajax.
b Ten minutes later, the forward had his second goal noted.

In this example, the phrases *the forward* and *his* in (2)b will be regarded as object-given, and therefore deaccented, because they refer to an entity (Kluivert) which was referred to earlier in the same paragraph (i.e., in the preceding sentence). Note that the example shows that object-givenness does not depend on the surface form of the referring expression, but only on its referent.

The second kind of givenness, concept-givenness, occurs if the root of a word has the same denotation as the root of a preceding word in the local discourse domain, or if the concept expressed by the second word subsumes the concept expressed by the first word. Sentence (2)b contains two instances of the first case: the words *minutes* and *goal* are regarded as concept-given due to the presence in the preceding sentence of the words *minute* and *goal* respectively.

Although the strategy of deaccenting given information usually produces correct accentuation patterns, in some cases too many words are deaccented. Using only the given/new distinction as a basis for accentuation may lead to accentuation patterns like the following:

(3) a After three minutes, *Feyenoord* took the lead through a goal by *Koeman*.
b In the sixth minute, *Kluivert* kicked a penalty home for *Ajax*.
c Ten minutes later, *Larsson* scored for Feyenoord.

These three sentences were all generated as part of the same paragraph. In (3)c, the word *Feyenoord* is deaccented due to givenness, because of the previous mention of *Feyenoord* in (3)a. This wrongly creates the impression that Kluivert scored for Feyenoord, just like Larsson. We see that the generated accentuation pattern does not fit together with the meaning of the sentence. To remedy this, *Feyenoord* should receive contrastive accent, indicating its contrast to *Ajax* in (3)b.

---

[3] Originally, this and the following examples of generated sequences are in Dutch. Since English and Dutch behave in a similar fashion with respect to accentuation I only show the English translations of the original sentences.

Examples like (3) illustrate what was already suggested by Chafe (1974), and - more recently - by Hirschberg (1992), van Deemter (1994) and Prevost (1995), namely that the given/new distinction is not sufficient to make predictions about accent: it is also necessary to distinguish contrastive accent. In order to generate the correct accentuation patterns for sentences like (3)c, the accentuation rules of GoalGetter should therefore be augmented with an algorithm for the assignment of contrastive accent. This means that the system must be able to recognize contrastive information, which is not a trivial problem. Before I describe the practical solution I implemented in GoalGetter, I will first discuss some theories on the prediction of contrastive accent.

2 Approaches to contrastive accent

In this section I will give a short and informal overview of three different approaches to the prediction of contrast, and point out their disadvantages. The discussion will be restricted to examples involving two subsequent sentences. The three approaches to contrast that I will discuss were proposed by Prevost (1995), van Deemter (1995) and Pulman (1997). They make use of alternative sets, parallelism and contrariety, and higher order unification respectively.

The theory of contrast proposed by Prevost (1995) was inspired by the 'alternative semantics' of Rooth (1992).[4] In Prevost's approach, an item receives contrastive accent if it co-occurs with another item that belongs to its 'set of alternatives', i.e., a set of different items of the same type. Prevost actually implemented his theory in a small generator, which can produce the responses in discourses like the following:

(4) Q: I know the American amplifier produces muddy treble,
but what kind of treble does the British amplifier produce?
A: The *British* amplifier produces *clean* treble

In the example, the two amplifiers are in each other's alternative sets, and so are the two kinds of treble. Because of the presence in the question of *American* and *muddy*, in the answer contrastive accent is assigned to *British* and *clean*.

There are two main problems with this approach. First, as Prevost himself notes, it is difficult to define exactly which items count as being of 'the same type'. If the definition is too strict, not all cases of contrast will be accounted for. On the other hand, if it is too broad, then anything will be predicted to contrast with anything. Prevost gives the following problematic example:

(5) While *he* intently watched the *clock*, *she* watched the *game*.

---

[4] Although Rooth deals with contrastive accent as well, I will not discuss his theory because it is purely aimed at the *interpretation* of focus (including contrastive accent), not its prediction.

This is a clear case of contrast, but it does not seem appropriate to regard *clock* and *game* as alternatives of each other, since they do not obviously share the same type. Allowing them to count as alternatives would mean an unwanted broadening of the notion of 'alternative set'.

A second problem is that there are cases where there is a clear co-occurrence of items of the same type, but no contrast, as in the following example from the football domain:

(6) a After three minutes, *Feyenoord* took the lead through a goal by *Koeman*.
b This caused *Ajax* to fall behind.
c Ten minutes later *Larsson* scored for Feyenoord.

Prevost's theory would predict *Feyenoord* in (6)c to have a contrastive accent, because the two teams Ajax and Feyenoord are obviously in each other's alternative set. In fact, though, *Feyenoord* should be normally deaccented due to givenness. This shows that the presence of an alternative item does not always trigger contrastive accent.

In the approach proposed by van Deemter (1995), contrast is accounted for in terms of parallelism and contrariety. The cases of contrast discussed above can be easily explained through a notion of parallelism which is closely linked to syntax (see, for instance, the proposal in Prüst (1992)). Both (4) and (5) show a clear parallelism between the succeeding sentences or clauses, while the absence of contrastive accent on *Feyenoord* in (6)c can be explained through a lack of parallelism between (6)b and (6)c.

Still, there are many examples of contrast which seem to lack parallelism. Van Deemter uses the notion of contrariety to account for these cases. Informally defined, two sentences (or clauses) are contrary to each other if they cannot be true at the same time. If two sentences contain two items which are 'contrastible' and whose substitution by the same constant will cause the sentences to be contrary to each other, then these sentences are said to stand in a contrast relationship and the contrastible items will receive contrastive accent. Inequality of denotations is the only condition determining whether two items are contrastible.

Van Deemter gives (7) as an example. If we assume that being an organ mechanic implies knowing much about organs, as stated in the meaning postulate (8), then replacing Mozart by Bach will result in a contrariety. This correctly predicts a contrastive accent on Bach and Mozart.

(7) *Bach* was an organ mechanic; *Mozart* knew little about organs
'Bach was an organ mechanic; Bach knew little about organs'

(8) $\forall x[\text{organ\_mechanic}(x)] \Rightarrow [\text{know\_much\_about\_organs}(x)]$

According to van Deemter, contrastive accent will also fall on those items which, after replacing them by the same constant, cause two sentences to be logically equivalent, as shown in (9).

(9) *Seven* is a prime number and so is *thirteen*
'Seven is a prime number and so is seven'

Apart from the fact that it is not immediately clear how this approach could be implemented in a generation system - checking for contrarieties would certainly require an impossible amount of world knowledge - there is a more important problem with this theory. Van Deemter's condition for 'contrastible items' is extremely permissive, allowing him to avoid the problems which Prevost encounters with examples like (5).[5] However, this liberal notion of contrastibility forces van Deemter to use a severe restriction on what counts as contrast: contrarieties (or equivalences) which are reached through more than one substitution do not qualify for contrastive stress, because otherwise far too many cases of contrastive stress would be predicted. Any pair of sentences of the form (NP$_1$ VP$_1$), (NP$_2$ Negation VP$_2$) or (NP$_1$ VP$_1$), (NP$_2$ VP$_2$) would then always count as contrastive, since substitution by the same constant of the NP's and of the VP's at the same time would lead to a contrariety or equivalence.

However, many examples of contrastive accentuation can only be explained if at least two pairs of items are substituted, because substitution of only one pair does not lead to a contrariety or equivalence. These cases cannot be accounted for by the theory. An example from the football domain is (10), where an equivalence (cf. (11)) can only be reached if the pairs *Koeman* - *Kluivert* and *fifth* - *twelfth* are substituted by a constant.

(10) In the *fifth* minute, the referee handed *Koeman* a yellow card; *Kluivert* received a yellow card in the *twelfth* minute
'In the fifth minute, the referee handed Koeman a yellow card; Koeman received a yellow card in the fifth minute'

(11) $\forall x[\text{referee\_hand\_card\_to}(x)] \Leftrightarrow [\text{receive\_card}(x)]$

The examples (7) and (10) can both be explained by Prevost's alternative set theory.

Another approach to the generation of contrastive accent is advocated by Pulman (1997), who proposes to use higher order unification (HOU) for the interpretation and prediction of focus, including contrastive accent. (See also Gardent and Kohlhase (1996) and Gardent et al. (1996).) Pulman makes use of equivalences like the following, which can be used for both interpretation and prediction of focus, and which operate at the level of quasi-logical forms or QLFs (Alshawi and Crouch (1992)):

(12) $\text{assert}(F,S) \Leftrightarrow S$
if
$B(F) = S$
& $\text{context}(C)$
& $P(A) = C$
& $\text{parallel}(B \bullet F, P \bullet A)$

---

[5] Although this particular example could be explained through parallelism in van Deemter's theory, there are other similar examples which do not show parallelism, e.g., 'While the *clock* was all *he* was paying attention to, *she* was watching the *game*'

This says that the QLF S with focus on F is equivalent to S if there is some sentence in the context with a QLF C, where C contains an item A that is parallel to F, while the background P of C (i.e., C after abstracting over A) is parallel to background B of S. Pulman does not define exactly when two items are parallel. Using HOU, equivalence (12) can be resolved in order to predict the landing place of focus markers in a generated sentence, with S being the QLF of this sentence. Pulman illustrates this with the following example, which I have simplified somewhat.

In the context of a system which generates information about the operation of some machinery, a user might ask *Do I put the card into the slot?*, which would be analysed as (13). Assuming that the correct operation at this point actually is that you put a disc into the slot, the semantics of the response by the system will be as given in (14).

(13) $\exists xy[put(user,x,y) \,\&\, card(x) \,\&\, slot(y)]$

(14) $\exists xy[put(user,x,y) \,\&\, disc(x) \,\&\, slot(y)]$

Now the equivalence in (12) will be resolved as follows. S is the QLF of the sentence to be generated, represented in (14). C will be equated to (13), where P = B = $\lambda \mathbf{P} \exists \mathbf{xy}[\mathbf{put(user,x,y)} \,\&\, \mathbf{P(x)} \,\&\, \mathbf{slot(y)}]$. A = **card** and F = **disc**. This means that the surface expression generated for **disc** should be marked for focus (in this case, contrastive accent).

Like van Deemter (1995), Pulman makes crucial use of parallelism, a notion which is as difficult to define as Prevost's alternative set. Pulman does not give a full definition of which items count as being parallel, but states that "to be parallel, two items need to be at least of the same type and have the same sortal properties" (Pulman (1997), p. 90). This condition is rather similar to Prevost's conditions on alternative sets. Consequently, Pulman's theory faces the same problem as Prevost's, namely that of defining when two items are of the same type. Like Prevost, Pulman can only explain the contrast in example (5) if *clock* and *game* count as being parallel, something which is not obvious.

Pulman has a theoretical advantage over Prevost in that he stresses that two sentences should not only contain some parallel items to warrant contrastive stress, as in Prevost (1995), but that the 'background' parts of the sentences should be parallel as well. In principle, this more restrictive condition on contrastive accent makes it possible for Pulman to account for examples like (6), which Prevost cannot explain: presumably, Pulman would not regard the backgrounds of (6)b and c as parallel.[6] However, as long as Pulman does not give a proper definition of parallelism, it is impossible to say what his theory will or won't predict.

As Gardent et al. (1996) point out, a HOU approach can take world knowledge into account when solving equations as in the example given above. They do not give an explicit description of *how* world knowledge can be used in solving equivalences, but presumably it could be done by making use of meaning postulates like those in (8) and (11) to solve those cases where the semantic representations of two sentences do not unify. For example, the contrast between the two clauses of (10) can be predicted if C in (12) is not equated to the direct semantic representation of the first clause, but to its equivalent according to (11). In this way (assuming a proper definition of parallelism is available), Pulman should be able to make the correct predictions for both (7) and (10). A similar solution might be possible for van Deemter (1995): by taking entailments and equivalences into account for the determination of parallelism, examples like (10), but also (7) could be accounted for in terms of parallelism. This way, checking for contrariety might become unnecessary.

---

[6] Prevost (personal communication) claims that he also looks for semantic parallelism between sentences, but this is not apparent from Prevost (1995).

To conclude, we have seen that a notion of semantic parallelism in combination with world knowledge seems to make the best predictions of contrast. However, a good definition of parallelism is lacking, and the encoding of world knowledge is a notorious problem. Even in a small domain like football reports the explicit enumeration of all possible semantic entailments and equivalences seems hardly feasible. Fortunately, data-to-speech systems like GoalGetter, the input of which is formed by typed and structured data, offer a simple way of automatically establishing semantic parallelism, with no need to explicitly encode world knowledge. In the next section, I will discuss how this can be done.

3 Contrastive accent in a data-to-speech system

The method I propose, and which has been successfully implemented in GoalGetter, is based on the simple principle that two sentences which express the same type of data structure (and therefore express similar information) should be regarded as contrastive. Contrastive accent should be assigned to those parts of the second sentence that express values which differ from those in the data structure expressed by the first sentence.

The idea behind this is the following. As we saw in the preceding section, for establishing contrast it is not sufficient to directly compare the semantic representations of two sentences: we need to use world knowledge to establish whether the sentences are semantically parallel, i.e. whether they describe similar situations or events. In our system this 'real world' information is readily available in the form of the data structures that are expressed by the sentences. We may consider two sentences semantically parallel if they express information contained in data structures of the same type, without caring about the specific linguistic forms chosen to convey this information. In this way, we can avoid the problems encountered by most of the theories discussed in section 2, as I will show in the rest of this section.

I will use example (3) from section 1 as an illustration. As was explained in that section, GoalGetter's football reports are generated on the basis of a typed data structure which is derived from the information on a Teletext page. The field `goallist` of this data structure contains a sequence of records of type *goal_event*, each record specifying the team for which a goal was scored, the player who scored, the time and the kind of goal: normal, own goal or a goal resulting from a penalty. The last two sentences of example (3) both express such a *goal_event* data structure, given in Figure 2, so they are regarded as contrastive, even though they show no direct syntactic or semantic parallelism.

$$\text{goal\_event (3)b} \quad \begin{bmatrix} \text{team:} & \text{Ajax} \\ \text{player:} & \text{Kluivert} \\ \text{minute:} & 6 \\ \text{goaltype:} & \text{penalty} \end{bmatrix}$$

$$\text{goal\_event (3)c} \quad \begin{bmatrix} \text{team:} & \text{Feyenoord} \\ \text{player:} & \text{Larsson} \\ \text{minute:} & 16 \\ \text{goaltype:} & \text{normal} \end{bmatrix}$$

Figure 2: Data structures expressed by (3)b and (3)c.

As can be seen in Figure 2, all the fields of the *goal_event* record expressed by (3)c have different values from that of (3)b. This means that all phrases in (3)c expressing the values of those fields should receive contrastive accent, including *Feyenoord*, despite its givenness. Note that the value of the `goaltype` field is not expressed in the surface structure of (3)c; however, if it were, it would receive a contrastive accent (e.g., *Ten minutes later, Larsson scored a* normal *goal for Feyenoord*).

Another example where lack of contrastive accent in GoalGetter used to lead to an incorrect accentuation pattern is the following sequence. Using only the given/new distinction without contrastive accent would lead to the following accentuation pattern:

(15) a In the sixteenth minute, the *Ajax* player *Kluivert* kicked the ball into the wrong goal.

b Twenty minutes later, *Overmars* scored for Ajax.

The deaccentuation of *Ajax* in (15)b gives the impression that both Kluivert and Overmars scored for Ajax, while in fact Kluivert scored for the other team through an own goal. Therefore, the second occurrence of Ajax should receive a contrastive accent despite its being given. In the theory of Prevost, this cannot be explained: (15)a does not contain a member of the alternative set of Ajax, so no contrast is predicted. Van Deemter's theory does not predict contrastive accent either, because (15)a and b do not show any parallelism, and contrariety only occurs after substitution of *two* pairs of items, the players and the times. Using Pulman's approach, contrast can only be predicted if the system contains the world knowledge that scoring an own goal means scoring for the opposing team.

The method proposed here does not require additional world knowledge to determine the presence of contrast in (15)b; the contrast can be immediately derived from the data structures expressed by sentences (15)a and b, which are given in Figure 3. A simple comparison of the `team` fields of (15)a and b shows that they have contrasting values, and that the phrase expressing the `team` field in (15)b should receive contrast accent, even though the corresponding value of the previous sentence was not overtly expressed.

$$\text{goal\_event (15)a} \begin{bmatrix} \text{team:} & \text{Feyenoord} \\ \text{player:} & \text{Kluivert} \\ \text{minute:} & \text{16} \\ \text{goaltype:} & \text{own} \end{bmatrix}$$

$$\text{goal\_event (15)b} \begin{bmatrix} \text{team:} & \text{Ajax} \\ \text{player:} & \text{Overmars} \\ \text{minute:} & \text{36} \\ \text{goaltype:} & \text{normal} \end{bmatrix}$$

Figure 3: Data structures expressed by (15)a and (15)b.

As we see, one of the advantages of the approach sketched above is that it requires no explicit listing of semantic equivalences or entailments. Only the information (data) which is expressed by a sentence is taken into account for the detection of contrast; which surface form is chosen to express certain information is not important. The discussion of examples (3) and (15) has shown that data can be expressed in an indirect way without influencing the prediction of contrast for the following sentence.

The approach sketched above will also give the desired result for example (6): sentence (6)c will not be regarded as contrastive with (6)b, since (6)c expresses a *goal_event* but (6)b does not. Therefore no contrastive accent will be assigned to *Feyenoord* in (6)c.

The approach can be extended to deal with deaccenting as well. Those parts of a sentence that express values which are identical to values in the data structure from which the previous sentence was generated, should be deaccented. This way, we can account for cases of deaccenting that cannot be handled by GoalGetter's current defocusing strategy, described in section 1. This can be illustrated by example (16), a variant of (10). The corresponding data structures are given in Figure 4. These structures are of type *card_event*, and describe at which time which player received a card of which colour.

(16) a In the *fifth* minute, *Koeman* was sent off the field.
b *Kluivert* received a red card in the *twelfth* minute.

$$\text{card event (16)a} \begin{bmatrix} \text{player:} & \text{Koeman} \\ \text{minute:} & \text{5} \\ \text{cardtype:} & \text{red} \end{bmatrix}$$

$$\text{card event (16)b} \begin{bmatrix} \text{player:} & \text{Kluivert} \\ \text{minute:} & \text{12} \\ \text{cardtype:} & \text{red} \end{bmatrix}$$

Figure 4: Data structures expessed by (16)a and (16)b.

Sentence (16)a expresses its underlying data in an implicit manner, leaving the colour of the card unspecified but inferrable. Sentence (16)b does explicitly mention the colour. Because the kind of card in this sentence is the same as in (16)a, the phrase expressing it (*red card*) is deaccented. This is not predicted by the defocusing strategy described in section 1, since in (16)a the type of card is not explicitly mentioned, and is therefore not detected by the defocusing algorithm. However, by looking at the data structures of (16)a and b, we can see that the values of the card feature are identical. The phrase *red card* in (16)b should therefore be deaccented. The result is the correct accentuation pattern as shown in (16), which will confirm the inference of the hearer that Koeman was shown a red card too.

Obviously, the proposed method places a great responsibility on the data structures that are used. The problem of defining parallelism is shifted to the design of the data structures: they must be set up in such a way that parallel items get assigned identical data types. It is still an open question whether it would be possible to specify general conditions on data structures, which they should meet in order to be usable for establishing contrast. So far, it seems that any data structure which is a plausible representation of the relevant domain, and which is rich enough to reflect the relations between objects in this domain, should be usable. This is confirmed by the fact that the data structure of GoalGetter was not designed for the prediction of contrast, but still proved to be suitable for this purpose.

4 Future work

The next step will be to see if the method described in the previous section can also be applied in another system, namely the OVIS system which is currently being developed in the Priority Programme Language and Speech Technology of NWO, the Netherlands Organization for Scientific Research. The OVIS dialogue system will provide information about public transport in the Netherlands. There already exists a typed data structure for this system, which has been designed independently from language generation. If this structure turns out to be usable for deriving contrast relations, this will prove that the applicability of the proposed method is not limited to GoalGetter.

Additionally, the principle on which the proposed method is based has to be further refined. For example, an open question which still remains is at which level data structures should be compared. Figures 3 and 4 presented data structures of type *goal_event* and *card_event* respectively. Since these data structures are of different types, currently they are not predicted to be contrastive. However, both are subtypes of a more general *event* type, which has only the fields `team`, `player`, and `minute`. For this reason, *goal_event* and *card_event* might have to be considered as contrastive after all. Examples like (17) seem to point in this direction.

(17) a In the eleventh minute, Ajax took the lead through a goal by *Kluivert*.
b Shortly after the break, the referee handed *Koeman* a yellow card.
c Ten minutes later, *Kluivert* scored for the second time.

The fact that Kluivert can be accented in (17)c can only be explained if (17)c is potentially contrastive to (17)b; otherwise, the second mention of Kluivert would be deaccented due to givenness, like *Feyenoord* in (6)c.

How such cases should be dealt with, will be the subject of further research. In general, the possibility of contrast between types and their subtypes (not only of events, but also of objects) should be further investigated. Presumably, both domain and discourse context play an important role here.

5 Conclusions

In this paper I have shown how the strategy of deaccenting given information can lead to incorrect accentuation patterns if contrast is not taken into account. Contrastive information should receive an accent, even if it is given. Approaches to the prediction of contrast which have been proposed in the literature are not attractive as a basis for implementation. The approach proposed by Prevost (1995) does not take parallelism between sentences into account and therefore does not always make the correct predictions. The contrast theory of van Deemter (1995) is too restrictive and cannnot account for all cases. Pulman (1997) does not give a proper definition of parallelism, and like the theory of van Deemter (1995), it requires a large amount of world knowledge in order to make the right predictions. Since it would be impossible to encode all relevant world knowledge, another solution must be found.

As an alternative, I have proposed a practical method to the assignment of contrastive accent in data-to-speech systems. In contrast to the approaches advocated by Prevost, van Deemter and Pulman, this method does not require a universal definition of alternative or parallel items. Also, the fact that determination of contrast is based on the information content of sentences obviates the need for explicitly encoding world knowledge; we can make use of the world knowledge which is already incorporated in the design of the data structures that are to be expressed. The use of these data structures for the prediction of contrastive accent is based on a general principle, which should be applicable in any system that generates sentences from a typed data structure.

The proposed approach has been implemented in the GoalGetter system and will be implemented in the OVIS system in the near future.

References

Alshawi, H. and R. Crouch (1992). Monotonic semantic interpretation. In *Proceedings of the 30th Annual Meeting of the ACL*, pp. 32–39.

Brown, G. (1983). Prosodic structure and the given/new distinction. In D. R. Ladd and A. Cutler (Eds.), *Prosody: Models and Measurements*, pp. 67–77. Berlin: Springer Verlag.

Chafe, W. (1974). Language and consciousness. *Language 50*, 111–133.

Chafe, W. (1976). Givenness, contrastiveness, definiteness, subjects, topics and points of view. In C. N. Li (Ed.), *Subject and Topic*, pp. 25–55. New York:

Academic Press.

Dirksen, A. (1992). Accenting and deaccenting: a declarative approach. In *Proceedings of COLING 1992, Nantes, France*, pp. 865–869. IPO MS. 867.

Dirksen, A. and H. Quené (1993). Prosodic analysis: the next generation. In van Heuven and Pols (Eds.), *Analysis and Synthesis of Speech: Strategic Research Towards High-Quality Text-to-Speech Generation*, pp. 131–144. Berlin - New York: Mouton de Gruyter.

Gardent, C. and M. Kohlhase (1996). Focus and higher-order unification. In *Proceedings of COLING 1996*, pp. 430–435.

Gardent, C., M. Kohlhase, and N. van Leusen (1996). Corrections and higher-order unification. In *Proceedings of KONVENS, Bielefeld*, pp. 268–279.

Halliday, M. (1967). Notes on transitivity and theme in English. *Journal of linguistics 3*, 199–244.

Hirschberg, J. (1992). Using discourse context to guide pitch accent decisions in synthetic speech. In G. Bailly, C. Benoit, and T. Sawallis (Eds.), *Talking Machines: Theories, Models and Designs*, pp. 367–376. Elsevier Science Publishers B.V.

Klabbers, E. (1997). Speech output generation in GoalGetter. In K. van Deemter, J. Landsbergen, J. Odijk, and G. Veldhuijzen van Zanten (Eds.), *CLIN VII, papers from the seventh CLIN meeting*.

Klabbers, E., J. Odijk, J. de Pijper, and M. Theune (1996). GoalGetter: From Teletext to speech. *IPO Annual Progress Report 31*, 61–75.

Klabbers, E., J. Odijk, J. de Pijper, and M. Theune (1997). From data to speech: a generic approach. IPO MS 1202.

Prevost, S. (1995). *A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation*. Ph. D. thesis, University of Pennsylvania.

Prüst, H. (1992). *On discourse structuring, VP anaphora and gapping*. Ph. D. thesis, University of Amsterdam.

Pulman, S. (1997). Higher order unification and the interpretation of focus. *Linguistics and Philosophy 20*, 73–115.

Rooth, M. (1992). A theory of focus interpretation. *Natural Language Semantics 1*, 75–116.

Theune, M., E. Klabbers, J. Odijk, and J. de Pijper (1997). Computing prosodic properties in a data-to-speech system. In *Proceedings of the Workshop on Concept-toSpeech Generation Systems, ACL/EACL 1997*, Madrid, pp. 39–45.

van Deemter, K. (1994). What's new? A semantic perspective on sentence accent. *Journal of Semantics 11*, 1–31.

van Deemter, K. (1995). Contrastive stress, contrariety and focus. In P. Bosch and R. v.d. Sandt (Eds.), *Focus & Natural Language Processing*. Cambridge: Cambridge University Press.

On the notion 'Minor Category'

Frank van Eynde[*]

Abstract

This paper presents an HPSG based treatment of minor signs, i.e. words which cannot head a phrasal projection. In contrast to what is commonly assumed in PSG, I will argue that the minor signs do not belong to separate speech parts, but that all speech parts have both major and minor members. This claim is substantiated with evidence from the Dutch personal pronouns and the English determiners. The consequences for the HPSG sort hierarchy are spelled out and a number of criteria are presented for identifying minor signs.

Introduction

Many syntactic frameworks make a distinction between major and minor categories. The definitions of the distinction do not always excel in clarity, but an account which is both clear and reasonably close to a theory-neutral understanding of the terms is the one of Generalized Phrase Structure Grammar. In Gazdar, Klein, Pullum, and Sag (1985) the distinguishing characteristic is that the members of major categories have a phrasal projection, whereas the members of minor categories do not. The former include the verbs, nouns, adjectives and prepositions, and these are the heads of resp. VPs, NPs, APs and PPs. The minor categories, on the other hand, include the complementizers, the coordinating conjunctions, the determiners and a number of degree words.[1]

MINOR	examples	
Complementizer	*that, for, if, whether*	p. 113
Conjunction	*and, or, nor, both, either, neither, but*	p. 171
Determiner	*the, a, this, that, which*	p. 126
Degree	*how, so, as, too, more, less*	p. 122

These categories do not have a phrasal projection, such as CompP or DetP; this reflects the fact that their members cannot take any syntactic dependents.

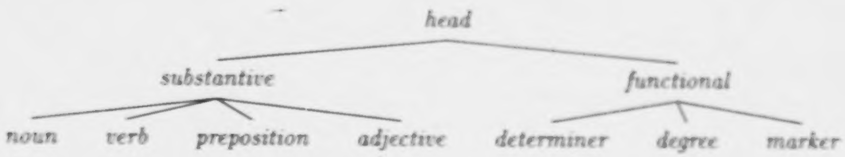
---

[*] Centrum voor Computerlinguistiek, K.U. Leuven. I would like to thank Valerio Allegranza, Ineke Schuurman and two anonymous reviewers for their comments on previous versions of this text.

[1] The page numbers in the last column refer to Gazdar, Klein, Pullum, and Sag (1985). The degree words *more* and *less* should be distinguished from the homonymous adjectives, cf. *more/less expensive* vs. *more/less wine*.

The GPSG treatment of minor categories has been criticised in Head-driven Phrase Structure Grammar. The main point of criticism concerns the status of the determiners and the degree words. In the analysis of Pollard and Sag (1994, 363-371) the determiner *more* in a phrase like *much more wine* is specified by *much*, which implies that determiners can take dependents and hence that they cannot be minor.[2] The same reasoning is applied to the degree word *as*, which is argued to be specified by *twice* in a phrase like *twice as productive*.

In order to accommodate these observations, HPSG makes a double distinction. On the one hand, it replaces the *major/minor* dichotomy with a distinction between substantive and functional speech parts, identifying the substantive ones with GPSG's major categories and the functional ones with GPSG's minor categories. On the other hand, it makes a further distinction within the functional speech parts between the elements with a phrasal projection (Det and Deg) and the ones without (Comp and Conj). As a generic name for the latter Pollard and Sag (1994) employs the term *marker*. The resulting speech part hierarchy looks as follows:



In spite of the differences in substance, the GPSG and HPSG treatments share the practice of making the distinction between major and minor categories in terms of speech parts. The main claim of this article now is that the distinction had better be treated as cross-categorial. The evidence for this claim will be based on an analysis of the Dutch personal pronouns and the English determiners.

1 Minor pronouns

The English personal pronouns do not take any complements, but this does not mean that they cannot have a phrasal projection, for most of them can take other kinds of dependents, such as adjectival modifiers, relative clauses or appositions:

(1) a. Poor me!
 b. Let he who is without sin throw the first stone.
 c. I, Benito Mussolino, challenge you.

As a consequence, these pronouns are major and have phrasal projections, just like the common nouns. In Dutch, however, we find a different situation, for in contrast to English, Dutch has two paradigms of personal pronouns: next to the one of the full pronouns, there is the paradigm of their reduced counterparts. The following survey is a summary of the data in Geerts, Haeseryn, de Rooij, and van den Toorn (1984, 163-167)[3]

---

[2] This criticism is not entirely justified, since Gazdar, Klein, Pullum, and Sag (1985, 126) treats words like *many*, *few* and their comparative and superlative counterparts as adjectives, rather than as determiners. It is true, though, that the degree word *more* can also be specified by *much*, as in *much more expensive*, and this is a word which GPSG does treat as minor.

person	number	gender	full-nom	full-acc	red-nom	red-acc
1st	sing	m/f	ik	mij	'k	me
	plur	m/f	wij	ons	we	
2nd	sing	m/f	jij	jou	je	je
	sg/pl	m/f	gij	u	ge	
3rd	sing	neut			het, 't	het, 't
	sing	masc	hij	hem	ie	'm
	sg/pl	fem	zij	haar	ze	ze, 'r, d'r
	plur	m/f/n	zij	hen, hun	ze	ze

Besides the fact that they cannot be stressed the reduced pronouns show a significant syntactic difference with the full pronouns: while the latter can be combined with a relative clause or an apposition, just like their English counterparts, the reduced pronouns cannot.

(2) Zij/*Ze die gaan sterven groeten u.
They who go die greet you.
'Those who are about to die greet you'

(3) Wij/*We, Albert, Koning der Belgen, ...
We, Albert, King of-the Belgians, ...
'We, Albert, King of the Belgians ...'

A related contrast is the one in *jij/*je daar* (= you there). As observed in Coppen (1991, 109), this use of the adverb *daar*, which intensifies the deictic meaning of the preceding nominal, is compatible with the full pronouns but not with the reduced ones.

Yet another relevant contrast is the one in

(4) Wij/*We mannen drinken graag bier.
We men drink willingly beer.
'we men like drinking beer'

In this case it is less obvious whether the head of the NP is the noun or the pronoun. Following the analysis which is proposed for *we sailors* in Postal (1969), it could be argued that the head of *wij mannen* is the noun and that the pronoun is its determiner, see also Jackendoff (1977, 106). However, what speaks against this analysis, is the fact that the person value of the subject is determined by the pronoun and not by the noun. In the case of a reflexive verb, like *zich vergissen*, for instance, the reflexive pronoun has to be of the first person, and not of the third, as would be normal for nonpronominal NPs, and as is in fact obligatory when the noun is combined with a possessive determiner:

---

[3] The table only mentions the nominative and accusative pronouns with reduced counterparts, this explains the absence of the second person plural *jullie* and the politeness form *u*, which have only got full forms. Notice the absence of full forms for the singular neuter *het*.

(5) Wij mannen vergissen ons/*zich zelden.
We men err ourselves/*themselves seldom.
'we men seldom err'

(6) Onze mannen vergissen zich/*ons zelden.
Our men err themselves/*ourselves seldom.
'our men seldom err'

This shows that the head of the NP had better be identified with the personal pronoun, and given the fact that the reduced pronouns cannot take any dependents, this is sufficient to account for the ungrammaticality of **we mannen*.

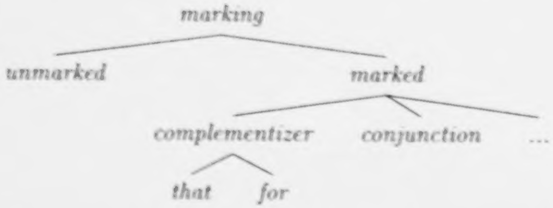
What these data suggest is that the full pronouns can take dependents and have phrasal projections, whereas their reduced counterparts cannot. Other differences between both types of pronouns will be discussed below, but first I will spell out the consequences of the distinction for the HPSG sort hierarchy.

2 *Major/Minor* as a cross-categorial distinction

In Pollard and Sag (1994) all signs have the same kind of CATEGORY value

$$ {}_{\textit{category}}\begin{bmatrix} \text{HEAD} & \text{head} \\ \text{SUBJ} & \text{list}\left(\text{synsem}\right) \\ \text{COMPS} & \text{list}\left(\text{synsem}\right) \\ \text{MARKING} & \text{marking} \end{bmatrix} $$

The HEAD value specifies the part of speech, together with some speech part specific information, such as case for nouns and verb form for verbs.[4] SUBJ and COMPS are valence features; they specify how many and what kind of subjects and/or complements a sign requires to be saturated. The MARKING feature is added for the elements which do not head a phrasal projection, i.e. the markers. Its possible values are



The markers get one of the subsorts of *marked* as their MARKING value; all other words receive the value *unmarked*.

In terms of this sort hierarchy, it is not clear how the reduced pronouns should be analyzed. The most obvious choice would be to treat them as NOMINAL, but in that case it is difficult to see how they can be distinguished from the full pronouns, for they will both be nominal and specified for case, they will both have empty lists for SUBJ and COMPS, and their MARKING values will systematically be *unmarked*. Furthermore, since PSG assumes that nouns are heads of nominal projections, it fails to capture the distinguishing characteristic of the reduced pronouns.

---

[4] For a survey of the speech part values, see the sort hierarchy in the introduction.

A second possibility would be to treat them as MARKERS, for that is the speech part to which the elements without phrasal projection belong. However, this implies that they cannot be nominal, and in that case it is not clear why they should show variation with respect to case. Moreover, since Pollard and Sag (1994) requires the complement daughters to be *phrasal*, it would follow that the reduced pronouns cannot be used as complements, and this is hard to square with the fact that their syntactic function is the same as that of the full pronouns.

A third possibility would be to claim that the reduced pronouns do not belong to any specific speech part, but that they are AFFIXES instead. This is not in conformity with their usual analysis in Dutch grammar, but it would be in line with the way in which Phrase Structure Grammar treats the clitic pronouns of the Romance languages, cf. Miller (1992) for French and Monachesi (1995) for Italian. In order to check whether the affix treatment would make sense for Dutch, let us briefly compare the Dutch reduced pronouns with the French clitics[5]

person	number	gender	full	cl-nom	cl-acc
1st	sing	m/f	moi	je, j'	me, m'
2nd	sing	m/f	toi	tu	te, t'
3rd	sing	masc	lui	il	le, l'
	sing	fem	elle		la, l'
	plur	masc	eux	ils	les
	plur	fem	elles		les

Like the Dutch reduced pronouns, the French clitics cannot take any syntactic dependents: in combination with an adjective or a relative clause, one has to use the full forms[6]

(7) Moi/*Je seule connais mon appétit.
I alone know my appetite.

(8) Lui/*Il qui était perdu est retrouvé.
He who was lost is found back.

Given this similarity it could be argued that the Dutch reduced pronouns had better be treated as affixes as well. Looking closer, though, it turns out that there are also some important differences. For a start, while the French clitics can only be complements of verbs, the Dutch reduced pronouns can also be complements of predicative adjectives and prepositions[7]

---

[5] The table does not mention the pronouns which lack a separate clitic form, such as the first and second person plural and the 'dative' pronouns *lui* and *leur*. Notice that the case distinction is only relevant for the clitic pronouns.

[6] A counterexample is the formulaic *Je soussigné, Pierre Lefèvre, déclare que ....* In Grevisse and Goosse (1989, 201) it is characterized as "un reste d'un ancien usage".

[7] The only minor pronoun which cannot be used as the complement of a preposition is *het*; in its place Dutch employs the -equally minor- *er*. This pronoun has to precede the preposition.

(9) Hij is de situatie/hen/het beu.
He is the situation/them/it fed up.
'He is fed up with the situation/them/it'

(10) Ik heb vannacht van jou/je gedroomd.
I have tonight of you dreamt.
'I've dreamt of you tonight'

In French, on the other hand, none of the clitic pronouns can be used as the complement of an adjective or a preposition, cf. *avec moi/*me* (= with me).

Another difference concerns the position of the pronouns. Whereas the French clitics must occur in the immediate vicinity of their head, the Dutch reduced pronouns can be separated from their heads by one or more constituents:

(11) ... dat ze me/je morgen eindelijk betalen.
... that they me/you tomorrow finally pay.
'... that they will finally pay me/you tomorrow'

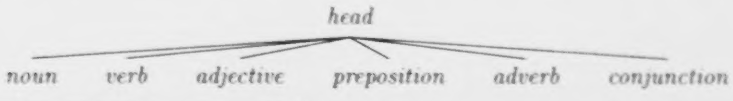
(12) We zijn het/ze eigenlijk al jaren beu.
We are it/them actually already for years fed up.
'Actually, we have been fed up with it/them for years now'

(13) Hij droomt er nu al jaren van.
He dreams it now already for years of.
'He has been dreaming of it for years now'

In each of these sentences there are two adjuncts in between the pronoun and its head, and more could be added. In sum, it appears that the Dutch reduced pronouns can be followed or preceded by virtually any kind of speech part, and this makes an affix based treatment highly implausible.

So far, we have considered three different ways of integrating the Dutch reduced pronouns in the standard HPSG sort hierarchy (noun, marker or affix), and none of them turns out to be satisfactory. Weighing their pros and cons, the least implausible is the first one, but it is also the one which fails to make the very distinction which we want to express. What is needed, apparently, is the possibility to treat the reduced pronouns as minor members of a 'major' speech part. In other words, we should foresee that the class of nouns does not only have major members, but also minor ones.

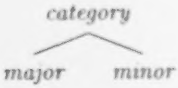
In order to enable this I will remove the distinction between elements with and without phrasal projection from the speech part hierarchy. In practice, this amounts to the cancellation of *marker* as a separate speech part[8]



---

[8] Anticipating the result of the discussion on minor determiners, I have also removed the value 'determiner' from the speech part hierarchy, so that the distinction between functional and substantive speech parts loses its relevance as well.

At the same time, I will apply the *major/minor* distinction to the objects of type *category*:



One consequence of this reshuffling is that the two classifications are mutually independent, and hence that every speech part may contain both major and minor members. Another consequence is that the feature declarations of major and minor signs can be differentiated. Exploiting this possibility, I will assume that all objects of type *category* have HEAD and MARKING features, but that only the ones of type *major* have got valence features.[9]

$$ {}_{\textit{category}}\begin{bmatrix} \text{HEAD} & \textit{head} \\ \text{MARKING} & \textit{marking} \end{bmatrix} \qquad {}_{\textit{major}}\begin{bmatrix} \text{SUBJ} & \textit{list}\left(\textit{synsem}\right) \\ \text{COMPS} & \textit{list}\left(\textit{synsem}\right) \end{bmatrix} $$

Making use of this modified hierarchy, the distinction between the major and the minor personal pronouns can be made explicit as follows:

$$ {}_{\textit{major}}\begin{bmatrix} \text{HEAD} & {}_{\textit{noun}}\begin{bmatrix} \text{CASE} & \textit{case} \end{bmatrix} \\ \text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \qquad {}_{\textit{minor}}\begin{bmatrix} \text{HEAD} & {}_{\textit{noun}}\begin{bmatrix} \text{CASE} & \textit{case} \end{bmatrix} \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} $$

Both types of pronouns are nominal and specified for case; the differences concern the presence of the valence features and the type of the CATEGORY value.

Besides the modifications to the speech part hierarchy we also need a relaxation of the constraints on nonhead daughters in phrasal signs. In Pollard and Sag (1994) the only nonhead daughters which are allowed to be words are the conjunction daughters and the marker daughters.[10] All other nonhead daughters are required to be phrasal. From what has been said so far, though, it is clear that this constraint is too strict, for the minor pronouns are nonphrasal but can be used as complement daughters nonetheless. For this reason I will relax the constraint that complement daughters have to be phrases and replace it with the more general requirement that they be signs[11]

---

[9] As an alternative, one could also claim that the minor elements have a COMPS list which is invariably empty. A possible advantage of this alternative is that it would simplify the definition of the notion 'nonhead daughter', for if minor elements do not have a COMPS list, the nonhead daughters have to be defined disjunctively, as either major signs with an empty COMPS list or minor signs, whereas if they have a COMPS list, the notion can be defined more succinctly as a sign with an empty COMPS list.

[10] The notion 'marker daughter' should be distinguished from the notion 'marker'. While the latter is the name of a speech part and hence contrasts with notions like 'noun' and 'verb', the former is the name of a syntactic function and contrasts with notions like 'head daughter' and 'complement daughter'. The difference between both notions is especially clear in the case of the coordinating conjunctions, for these are markers, but not marker daughters.

headed-phrase ⇒ [COMP-DTRS list (sign)]

Interestingly, this relaxation is not just needed for the treatment of minor pronouns, it also facilitates the elimination of vacuous projection from the grammar. For, if complements have to be phrasal, then one needs special measures to allow for one-word complements, as in *find John/him/gold/coins*, whereas in a treatment which allows complements to be single words, there is no need for any special measures.[12]

3 On the syntax of minor signs

So far, the minor signs have been characterized as elements which cannot take any syntactic dependents. At this point, with the new sort hierarchy in place, this property can be spelled out in formal detail, and related to a number of further distinctions between major and minor signs. For a start, since minor signs cannot take any syntactic dependents, they do not have a phrasal projection, and this implies that all phrasal signs are major:

phrase ⇒ [SYNSEM|LOC|CAT *major*]

As HPSG foresees only two types of signs, i.e. words and phrases, this amounts to the claim that minor signs must be of type *word*.

Second, in order to express the defining property of the minor signs that they cannot head a phrasal projection, it is sufficient to require that in a headed phrase the *head daughter* have a CATEGORY value of type *major*:

headed-phrase ⇒ [HEAD-DTR|SYNSEM|LOC|CAT *major*]

Third, in nonheaded phrases there are some further constraints. In coordinate phrases, for instance, the *conjunct daughters* have to be major:

(14) Ik twijfel nog tussen Mark en jou/*je.
I hesitate still between Mark and you.
'I'm still hesitating between Mark and you'

(15) Ik weiger te onderhandelen met hen/*ze en hun aanhangers.
I refuse to negotiate with them and their allies.
'I refuse to negotiate with them and their allies'

Interestingly, this constraint does not have to be stipulated, since it follows from the COORDINATION PRINCIPLE, Pollard and Sag (1994, 203).

---

[11] Here and throughout the paper I follow the practice of Sag (to appear) to apply the distinction between constituent structure types to the objects of type *phrase*. As a consequence, instead of saying that some phrase has a DAUGHTERS value of type *headed-structure*, as in Pollard and Sag (1994), I simply say that the phrase itself is of type *headed-phrase*. As in the case of words, the more specific types inherit the feature declarations and constraints of their supertypes.

[12] The same remark applies to the subject, adjunct and specifier daughters. They all may consist of a single word, and will therefore be required to be signs, rather than phrases.

In a coordinate structure, the CATEGORY and NONLOCAL value of each conjunct daughter is subsumed by (is an extension of) that of the mother.

Since coordinate structures are by definition phrasal, they have CATEGORY values of type *major*, and given the principle this implies that the conjunct daughters cannot be minor. As applied to the English personal pronouns, this predicts that they can all be used as conjuncts, and this is indeed the case, even for the singular neuter *it*:

(16) Recently speculation has been growing that it and the Roman Catholic Church will reunite. (TIME, May 5th, 1997, p. 47)

(17) Seen 800 years later, it and the other works in this superb exhibition still amaze and inspire. (TIME, May 5th, 1997, p. 54)

This possibility does not exist for its Dutch equivalent *het*.

A corollary of the above constraints is that a phrase has to contain at least one major daughter, i.e. the head daughter in headed phrases or the conjunct daughters in coordinate phrases. Put in other words, this amounts to the claim that a minor sign must have at least one major sister. Further evidence for this general requirement is provided by the fact that the minor pronouns cannot be the sole constituents of elliptical clauses. In reduced answers, for instance, one has to use the major pronouns:

(18) Wie heeft het gedaan ? Zij/*Ze.
Who has it done ? She.
'Who did it ? She did'

(19) Wie hebben ze gekozen ? Jou/*Je.
Whom have they chosen ? You.
'Whom did they choose ? You'

If we make the reasonable assumption that elliptical clauses are phrasal, then the exclusion of the minor pronouns in this position follows from the fact that a phrase has to contain at least one major daughter. This also makes the right predictions in the case of elliptical comparative clauses:

(20) Hij heeft meer gereisd dan zij/*ze.
He has more traveled than she.
'He has traveled more than she has'

(21) Het zal langer duren dan zij/ze denkt.
It will longer take than she thinks.
'It'll take longer than she thinks'

In the first sentence the minor pronoun cannot be used since there are no other constituents in the comparative clause, but in the second sentence the use of minor *ze* is allowed, since there is another constituent which qualifies as major, i.e. the verb *denkt*.

In sum, phrases are major and must have at least one major daughter, or -put differently- minor signs are words and must have at least one major sister.

4 Minor determiners

In order to demonstrate that the criteria for identifying minor signs are sufficiently general to be applicable to other languages and to other speech parts, I will now discuss the English NP specifiers. As a starting point I will use the following survey:

Articles	*the, a(n).*
Demonstratives	*this, that, these, those.*
Possessives	*my, our; your; his, her, its, their.*
Wh-determiners	*which(ever), what(ever), whose(ver).*
Logical determiners	*every, some, any, no.*
Numerals	*one, two, three, ...*

This list includes most of the words which are usually treated as NP specifiers in English grammar.[13] Semantically, they can be divided in two classes: the quantifying ones, which include the numerals and the logical determiners, and the deictic or anaphoric ones, which include the possessives, the demonstratives and the *wh*-determiners. This semantic distinction corresponds to a syntactic one: if an NP contains a determiner of either kind, the deictic/anaphoric one invariably has to precede the quantifying one.[14]

D/A-Determiner	Q-Determiner	Nominal
that/my	one	green bottle
your/whose	two	sisters
these/which	five	tables
his	every	word

Not all combinations of determiners are allowed (cf. *his every word* vs. **his no word*), but if the combination is allowed, then the quantifying determiner has to follow the deictic/anaphoric one. What will be argued now is that both classes of determiners contain some minor members.

4.1 The quantifying determiners

Starting with the numerals, it is clear that they are major, for they can be specified by adverbs which express how the quantity of the nominal's denotation compares to the quantity which is denoted by the numeral, as in *almost fifty, exactly one, nearly twelve* and *at least five*. This major status is confirmed by the fact that they can be conjoined, as in *six or seven tables*. As for their speech part, many authors introduce a separate category, such as Numeral or Cardinal; this practice is also followed in Pollard and Sag (1994, 366), which employs the term *Scalar*, albeit only "for expository convenience". Other sources argue that the numerals can be grouped with other independently needed speech parts. Jackendoff (1977), for instance, claims that the numerals are either Nouns or Quantifiers, depending on their position in the noun phrase. The proposal which is most commonly adopted in the current literature, though, is to treat the (adnominal) numerals as adjectives, see a.o. McCawley (1981, 430), Hoeksema (1983), Link (1987) and Allegranza (to appear). The evidence for the adjectival treatment which these authors present is mainly of a semantic nature, but also from a strictly syntactic point of view this proposal makes good sense, first because the specifiers which the numerals can take are the same as the ones which can be used with such nongradable adjectives as *impossible*, *dead* and *indistinguishable*, and second because the numerals may be preceded by other adjectives, as in *the last/next three days*, *the same five cars* and *the only/other two objections I can think of now*.

---

[13] Not included are the ordinals and the gradable determiners *much, many, little* and *few* with their comparative and superlative counterparts. They are all major and hence irrelevant for the identification of minor signs.

[14] The D/A-determiners may be preceded by a so-called predeterminer, such as *all* or *both*, a fraction like *half* or a multiplier like *twice*, as in *all/both their children* and *half/twice that size*, see Quirk, Greenbaum, Leech, and Svartvik (1985, 257-261). These elements share the quantifying nature of the Q-determiners, but syntactically they behave rather differently. Notice, for instance, that they do not only combine with nominal projections, but also with verbal or adverbial ones, as in *they will all/both go to Rome* and *twice as long*.

What is interesting now is that the numerals can be shown to have a minor member, i.e. the indefinite article *a(n)*. Both in form and in meaning, it clearly resembles the singular numeral *one*,[15] but while the latter can be specified, conjoined and stranded in elliptical comparative clauses, the former cannot:

(22) a. There is exactly one/*a car in the street.
b. Do you want one/*an or two cards ?
c. Two horses can carry more than one/*a.

This suggests that the indefinite article is the minor counterpart of the numeral, and since there is no reason to assume that minor signs belong to another speech part than their major counterparts, it follows that the indefinite article is a minor adjective. Some further evidence for this adjectival status is provided by the fact that it can be preceded by other adjectives or APs, as in *many a friend*, *such a man* and *too tall a building*.

Integrating this analysis in the HPSG sort hierarchy, I will assume that the MOD(IFIED) value of the numerals specifies the kind of nominal with which (the phrasal projection of) the numeral combines. In the case of *one*, for instance, this is a singular count nominal:

$${}_{\textit{major}}\left[\begin{array}{ll} \text{HEAD} & {}_{\textit{adjective}}\left[\text{MOD} \quad \text{N}'\left[\text{sing, count}\right]\right] \\ \text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MARKING} & \textit{unmarked} \end{array}\right]$$

Having empty lists for SUBJ and COMPS, the numeral cannot take any complements or subjects, but being major, it can take specifiers, as in *at least one*, and it can be conjoined as in *one or two questions*; its phrasal projection is an adnominal adjunct, as in *at least one bike*. The indefinite article, on the other hand, has the same HEAD and MARKING values, but lacks the valence features and has another type of CATEGORY value. This is sufficient to make explicit that it cannot be used in any other way than as the specifier of a singular count noun. Still, there is one further difference: whereas the numerals can be preceded by another determiner, as in *that/the one bottle he threw away*, the indefinite article cannot: **that/the a bottle*. In order to capture this difference I will assume that the numeral combines with a nominal object and yields another nominal object, whereas the indefinite article combines with a nominal object and yields a quantifier.[16] This, together with the assumption that the D/A-determiners combine with a nominal object and yield a quantifier, is sufficient to make the required differentiation. In sum, the AVM of the indefinite article can be specified as follows:

---

[15] There are languages in which the indefinite article is even homonymous to the numeral, cf. the the German *ein*, the French *un* and the Italian *uno*.

$$\begin{bmatrix} \text{CAT} & {}_{\textit{minor}}\begin{bmatrix} \text{HEAD} & {}_{\textit{adjective}}\begin{bmatrix} \text{MOD} & \text{N}'\begin{bmatrix}\text{sing, count}\end{bmatrix}: \boxed{1} \end{bmatrix} \\ \text{MARKING} & \text{unmarked} \end{bmatrix} \\ \text{CONTENT} & {}_{\textit{quantifier}}\begin{bmatrix} \text{DET} & \text{exists} \\ \text{RESTIND} & \boxed{1}\ \text{nominal-object} \end{bmatrix} \end{bmatrix}$$

In this way all significant differences with the numeral *one* are captured without having to assume that the indefinite article belongs to another speech part.

Turning to the logical determiners, it is easy to find evidence for major status, for they can take roughly the same kinds of specifiers as the numerals (*almost every, at least some, virtually any* and *practically no*), and they can be used as conjuncts:

(23) a. Some but not all flowers are yellow.
b. There is little or no money left.
c. She was looking under each and every stone.

Just like the numerals, though, the logical determiners can be argued to contain a minor member as well, i.e. the unstressed *some*.[17]

(24) a. At least some/*sm problems have been solved.
b. Some/*Sm but not all pupils will be there.

With the exception of *every*, none of the logical determiners can be preceded by a D/A-determiner; this implies that they are of the same semantic type as the indefinite article, i.e. they combine with a nominal object and yield a quantifier. As for the speech part of the logical determiners, one finds various proposals, ranging from Quantifier over Determiner to Article. Within the present context, though, the most natural option is to assign them the same speech part as the numerals, first because they take the same kind of specifiers, and second because their minor members are in complementary distribution: the unstressed *sm* is typically used in those combinations in which the indefinite article cannot be used, i.e. with mass nouns and plural count nouns:

---

[16] The HPSG distinction between nominal object and quantifier is comparable to the distinction between a set and a set of sets in Generalized Quantifier Theory.

[17] In order to differentiate the stressed determiner from its minor counterpart, I will use *some* for the former and *sm* for the latter. From a cross-linguistic perspective, *sm* corresponds to the partitive articles of the Romance languages.

(25) a. Would you like sm/*a water ?

b. I'm going to buy sm/*a potatoes.

It can be concluded then that *sm* is a minor adjective as well.

4.2 The deictic and anaphoric determiners

As for the deictic or anaphoric determiners, the possessives are clearly major, for they can be conjoined and specified by the adverb *own*:

(26) a. Shall we take my or your car ?

b. Every country gives priority to its own interests.

For the demonstratives it is less clear what kind of specifiers they can take, but their major status is clear from the fact that they can be conjoined and stranded in an elliptical comparative clause:

(27) a. Shall we take this or that carpet ?

b. I like these apples better than those.

Besides these major members, the demonstratives can be argued to have a minor one as well, i.e. the definite article *the*. Both in form and meaning it resembles the demonstrative *that*,[18] but in contrast to the latter it cannot be conjoined nor stranded:

(28) a. * Shall we buy the or this carpet ?

b. * I like these apples better than the.

As for the speech part of the demonstratives, many authors postulate an *ad-hoc* category, such as Demonstrative or Article. Within the logic of the present treatment, though, it is more appropriate to put them in the same class as the quantifying determiners. Notice, for instance, that they share the property of the quantifying determiners to impose constraints on the number value of the head noun: *this* and *that* require the singular, just like *one* and *every*, whereas *these* and *those* require the plural, just like *two* and *three*. As a consequence, since the quantifying determiners have been argued to be adjectives, it follows that the demonstratives can best be treated as adjectival as well. Further evidence for this status is provided by the fact that the singular demonstratives share the property of a number of adjectives to have an adverbial homonym: adjectives like *pretty*, *wide* and *real*, for instance, have degree denoting homonyms, as in *a pretty difficult task*, *be wide awake* and *a real nice girl*. Such homonyms also exist for *this* and *that*, as in *this long* and *that short*; as a matter of fact, the definite article has a similar adverbial use in correlative constructions like *the sooner, the better*. In sum, it does not seem too far-fetched to assume that the English demonstratives are adjectives, and to treat the definite article as a minor adjective:

---

[18] In some languages, they are even homonymous. In German, for instance, the definite article has exactly the same paradigm of forms as the demonstrative *der/die/das*.

$$\begin{bmatrix} \text{CAT} & {}_{minor}\begin{bmatrix} \text{HEAD} & {}_{adjective}\begin{bmatrix} \text{MOD} & \text{N}' : \boxed{1} \end{bmatrix} \\ \text{MARKING} & \text{unmarked} \end{bmatrix} \\ \text{CONTENT} & {}_{quantifier}\begin{bmatrix} \text{DET} & \text{the} \\ \text{RESTIND} & \boxed{1}\ \text{nominal-object} \end{bmatrix} \end{bmatrix}$$

Because of the constraint on the CONTENT value of the head, the definite article cannot be combined with another D/A-determiner, nor with a Q-determiner which yields an object with CONTENT value of type *quantifier*, such as the indefinite article or the logical determiners.

Interestingly, these conclusions have some consequences for the much debated issue of whether the head of a noun phrase is the noun or the determiner (cf. NP vs. DP), see a.o. Abney (1986), Hudson (1990), Van Langendonck (1994) and –within HPSG– Pollard and Sag (1994, 363-371), Netter (1994, 301-305) and Allegranza (to appear). In this section the issue has not been addressed directly, but the fact that the determiners have been argued to be adjectives provides indirect evidence for the NP analysis, since it is commonly accepted that the head of an [Adj+Noun] combination is the noun rather than the adjective. Furthermore, since the articles and unstressed *sm* are minor, they cannot be head daughters, so that in combinations like *a dog, sm sugar* and *the cat* the head daughter must be the noun. In sum, while the main aim of this section was to provide evidence for the existence of minor determiners, we have also provided some indirect evidence for the assumption that [Det+Noun] combinations are headed by the noun.

5 Summing up

The main claim of this paper is that the distinction between major and minor signs should be treated as cross-categorial. The evidence for this claim is based on an analysis of the Dutch personal pronouns and the English determiners. Employing the criterion that the minor signs are words which cannot take any syntactic dependents I have shown that both of these classes contain some minor members[19]

	major	minor
noun	Dutch full pronouns	Dutch reduced pronouns
adjective	English numerals	indefinite article *a(n)*
	English logical determiners	unstressed *some*
	English demonstratives	definite article *the*

---

[19] This covers only two of the traditional parts of speech, but in other work I have shown that the distinction also applies to prepositions and to Dutch and German verbs, cf. Van Eynde (1994, 53-60,179-192).

As part of the argumentation, I have identified a number of further characteristics of the minor signs, i.e. the impossibility to be conjoined and to be stranded under ellipsis. Taken together, these constraints amount to the claim that a phrase must contain at least one major daughter, or –in other words– that a minor sign must have at least one major sister.

While this criterion is sufficiently general to be applicable to all languages and to all speech parts, it may be worth stressing that the result of its application is language specific. For example, when the criterion is applied to the personal pronouns, it turns out that the English ones are all major, whereas the Dutch ones can be divided in major and minor ones. Similarly, when applied to the NP specifiers, it turns out that English has both major and minor determiners, whereas languages without articles, such as Latin and Russian, have probably only got major determiners.

References

Abney, S. (1986). *The English Noun Phrase in its Sentential Aspect.* Ph. D. thesis, MIT.

Allegranza, V. (to appear). Determiners as Functors: NP Structure in Italian. In S. Balari and L. Dini (Eds.), *HPSG for Romance*. Stanford: CSLI Publications. (to appear).

Coppen, P. A. (1991). *Specifying the Noun Phrase.* Ph. D. thesis, Katholieke Universiteit Nijmegen.

Gazdar, G., E. Klein, G. Pullum, and I. Sag (1985). *Generalized Phrase Structure Grammar.* Oxford: Basil Blackwell.

Geerts, G., W. Haeseryn, J. de Rooij, and M. van den Toorn (Eds.) (1984). *Algemene Nederlandse Spraakkunst.* Groningen/Leuven: Wolters-Noordhoff.

Grevisse, M. and A. Goosse (1989). *Nouvelle grammaire française.* Paris/Louvain-la-Neuve: Duculot. Deuxième édition.

Hoeksema, J. (1983). Plurality and conjunction. In A. Ter Meulen (Ed.), *Studies in Modeltheoretic Semantics*, Number 1 in GRASS. Dordrecht: Foris.

Hudson, R. (1990). *English Word Grammar.* Oxford: Blackwell.

Jackendoff, R. S. (1977). *$\overline{X}$ Syntax: A Study of Phrase Structure.* Cambridge MA: MIT Press.

Link, G. (1987). Generalized quantifiers and plurals. In P. Gärdenfors (Ed.), *Generalized Quantifiers.* Dordrecht: Reidel.

McCawley, J. D. (1981). *Everything that Linguists have Always Wanted to Know about Logic (but were ashamed to ask).* University of Chicago Press.

Miller, P. H. (1992). *Clitics and Constituents in Phrase Structure Grammar.* New York: Garland. Published version of 1991 Doctoral dissertation, University of Utrecht, The Netherlands.

Monachesi, P. (1995). *A grammar of Italian clitics.* Ph. D. thesis, Tilburg University.

Netter, K. (1994). Towards a Theory of Functional Heads: German Nominal Phrases. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-Driven Phrase Structure Grammar*, pp. 297–340. Stanford: CSLI Publications.

Pollard, C. and I. Sag (1994). *Head-driven Phrase Structure Grammar.* Stanford/Chicago: CSLI and University of Chicago Press.

Postal, P. (1969). On so-called "Pronouns" in English. In D. Reibel and S. Schane (Eds.), *Modern Studies in English*, pp. 201–224. New Yersey: Englewood Cliffs.

Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik (1985). *A comprehensive grammar of the English language.* London/New York: Longman.

Sag, I. (to appear). English Relative Clause Constructions. *Journal of Linguistics*.

Van Eynde, F. (1994). *Auxiliaries and Verbal Affixes. A monostratal cross-linguistic analysis.* Habilitationsschrift, Katholieke Universiteit Leuven.

Van Langendonck, W. (1994). Determiners as heads? *Cognitive Linguistics 5*(3), 243–259.

Resolving PP attachment Ambiguities with Memory-Based Learning

Jakub Zavrel*
Walter Daelemans*
Jorn Veenstra*

Abstract

In this paper we describe the application of Memory-Based Learning to the problem of Prepositional Phrase attachment disambiguation. We compare Memory-Based Learning, which stores examples in memory and generalizes by using intelligent similarity metrics, with a number of recently proposed statistical methods that are well suited to large numbers of features. We evaluate our methods on a common benchmark dataset and show that our method compares favorably to previous methods, and is well-suited to incorporating various unconventional representations of word patterns such as value difference metrics and Lexical Space.

Introduction

A central issue in natural language analysis is structural ambiguity resolution. A sentence is structurally ambiguous when it can be assigned more than one syntactic structure. The drosophila of structural ambiguity resolution is Prepositional Phrase (PP) attachment. Several sources of information can be used to resolve PP attachment ambiguity. Psycholinguistic theories have resulted in disambiguation strategies which use syntactic information only, i.e. structural properties of the parse tree are used to choose between different attachment sites. Two principles based on syntactic information are Minimal Attachment (MA) and Late Closure (LC) (Frazier 1979). MA tries to construct the parse tree that has the fewest nodes, whereas LC tries to attach new constituents as low in the parse tree as possible. These strategies always choose the same attachment regardless of the lexical content of the sentence. This results in a wrong attachment in one of the following sentences:

1 *She eats pizza with a fork.*

2 *She eats pizza with anchovies.*

---

*ILK, Induction of Linguistic Knowledge, Tilburg University

In sentence 1, the PP "with a fork" is attached to the verb "eats" (high attachment). Sentence 2 differs only minimally from the first sentence; here, the PP "with anchovies" does not attach to the verb but to the NP "pizza" (low attachment). In languages like English and Dutch, in which there is very little overt case marking, syntactic information alone does not suffice to explain the difference in attachment sites between such sentences. The use of syntactic principles makes it necessary to re-analyse the sentence, using semantic or even pragmatic information, to reach the correct decision. In the example sentences 1 and 2, the meaning of the head of the object of 'with' determines low or high attachment. Several semantic criteria have been worked out to resolve structural ambiguities. However, pinning down the semantic properties of all the words is laborious and expensive, and is only feasible in a very restricted domain. The modeling of pragmatic inference seems to be even more difficult in a computational system.

Due to the difficulties with the modeling of semantic strategies for ambiguity resolution, an attractive alternative is to look at the statistics of word patterns in annotated corpora. In such a corpus, different kinds of information used to resolve attachment ambiguity are, implicitly, represented in co-occurrence regularities. Several statistical techniques can use this information in learning attachment ambiguity resolution.

Hindle and Rooth (1993) were the first to show that a corpus-based approach to PP attachment ambiguity resolution can lead to good results. For sentences with a *verb/noun* attachment ambiguity, they measured the lexical association between the *noun* and the *preposition*, and the *verb* and the *preposition* in unambiguous sentences. Their method bases attachment decisions on the ratio and reliability of these association strengths. Note that Hindle and Rooth did not include information about the second noun and therefore could not distinguish between sentence 1 and 2. Their method is also difficult to extend to more elaborate combinations of information sources.

More recently, a number of statistical methods better suited to larger numbers of features have been proposed for PP-attachment. Brill and Resnik (1994) applied Error-Driven Transformation-Based Learning, Ratnaparkhi, Reynar and Roukos (1994) applied a Maximum Entropy model, Franz (1996) used a Loglinear model, and Collins and Brooks (1995) obtained good results using a Back-Off model.

In this paper, we examine whether Memory-Based Learning (MBL), a family of statistical methods from the field of Machine Learning, can improve on the performance of previous approaches. Memory-Based Learning is described in Section 1. In order to make a fair comparison, we evaluated our methods on the common benchmark dataset first used in Ratnaparkhi, Reynar, and Roukos (1994). In section 2, the experiments with our method on this data are described. An important advantage of MBL is its use of *similarity-based reasoning*. This makes it suited to the use of various unconventional representations of word patterns (Section 1.3). In Section 2.2 a comparison is provided between two promising representational forms. Section 3 contains a comparison of our method to previous work, and we conclude with section 4.

1 Memory-Based Learning

Classification-based machine learning algorithms can be applied in learning disambiguation problems by providing them with a set of examples derived from an annotated corpus. Each example consists of an input vector representing the context of an attachment ambiguity in terms of features (e.g. syntactic features, words, or lexical features in the case of PP-attachment), and an output class (one of a finite number of possible attachment positions representing the correct attachment position for the input context). Machine learning algorithms extrapolate from the examples to new input cases, either by extracting regularities from the examples in the form of rules, decision trees, connection weights, or probabilities in greedy learning algorithms, or by a more direct use of analogy in lazy learning algorithms. It is the latter approach which we investigate in this paper. It is our experience that lazy learning (such as the Memory-Based Learning approach adopted here) is more effective for several language-processing problems (see Daelemans (1995) for an overview) than more eager learning approaches. Because language-processing tasks typically can only be described as a complex interaction of regularities, subregularities and (families of) exceptions, storing all empirical data as potentially useful in analogical extrapolation works better than extracting the main regularities and forgetting the individual examples (Daelemans 1996).

1.1 Analogy from Nearest Neighbors

The techniques used are variants and extensions of the classic k-nearest neighbor (k-NN) classifier algorithm. The instances of a task are stored in a table, together with the associated "correct" output. When a new pattern is processed, the k nearest neighbors of the pattern are retrieved from memory using some similarity metric. The output is determined by extrapolation from the k nearest neighbors. The most common extrapolation method is *majority voting* which simply chooses the most common class among the k nearest neighbors as an output.

1.2 Similarity metrics

The most basic metric for patterns with symbolic features is the **Overlap metric** given in Equations 1 and 2; where $\Delta(X,Y)$ is the distance between patterns X and Y, represented by n features, w_i is a weight for feature i, and δ is the distance per feature. The k-NN algorithm with this metric, and equal weighting for all features is called IB1 (Aha, Kibler, and Albert 1991). Usually k is set to 1.

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \, \delta(x_i, y_i) \tag{1}$$

where:

$$\delta(x_i, y_i) = 0 \; if \; x_i = y_i, \; else \; 1 \tag{2}$$

This metric simply counts the number of (mis)matching feature values in both patterns. If no information about the importance of features is available, this is a reasonable choice. But if we have information about feature relevance, we can add linguistic bias to weight or select different features (Cardie 1996). An alternative, more empiricist, approach is to look at the behavior of features in the set of examples used for training. We can compute statistics about the relevance of features by looking at which features are good predictors of the class labels. Information Theory provides a useful tool for measuring feature relevance in this way, see (Quinlan 1993).

Information Gain (IG) weighting looks at each feature in isolation, and measures how much information it contributes to our knowledge of the correct class label. The Information Gain of feature f is measured by computing the difference in uncertainty (i.e. entropy) between the situations without and with knowledge of the value of that feature (Equation 3):

$$w_f = \frac{H(C) - \sum_{v \in V_f} P(v) \times H(C|v)}{si(f)} \tag{3}$$

$$si(f) = -\sum_{v \in V_f} P(v) \log_2 P(v) \tag{4}$$

Where C is the set of class labels, V_f is the set of values for feature f, and $H(C) = -\sum_{c \in C} P(c) \log_2 P(c)$ is the entropy of the class labels. The probabilities are estimated from relative frequencies in the training set. The normalizing factor $si(f)$ (split info) is included to avoid a bias in favor of features with more values. It represents the amount of information needed to represent all values of the feature (Equation 4). The resulting IG values can then be used as weights in Equation 1. The k-NN algorithm with this metric is called IB1-IG (Daelemans and van den Bosch 1992).

The possibility of automatically determining the relevance of features implies that many different and possibly irrelevant features can be added to the feature set. This is a very convenient methodology if theory does not constrain the choice sufficiently beforehand, or if we wish to measure the importance of various information sources experimentally.

1.3 MVDM and LexSpace

Although IB1-IG solves the problem of feature relevance to a certain extent, it does not take into account that the symbols used as values in the input vector features (in this case words, syntactic categories, etc.) are not all equally similar to each other. According to the Overlap metric, the words *Japan* and *China* are as similar as *Japan* and *pizza*. We would like *Japan* and *China* to be more similar to each other than *Japan* and *pizza*. This linguistic knowledge could be encoded into the word representations by hand, e.g. by replacing words with semantic labels, but again we prefer a more empiricist approach in which distances between values of the same feature are computed differentially on the basis of properties of the training set. To this end, we use the Modified Value Difference Metric (MVDM) of Cost and Salzberg (1993); a variant of a metric first defined in Stanfill and Waltz (1986). This metric (Equation 5) computes the frequency distribution of each value of a feature over the categories. Depending on the similarity of their distributions, pairs of values are assigned a distance.

$$\delta(V_1, V_2) = \sum_{i=1}^{n} |P(C_i|V_1) - P(C_i|V_2)| \tag{5}$$

In this equation, V_1 and V_2 are two possible values for feature f; the distance is the sum over all n categories; and $P(C_i|V_j)$ is estimated by the relative frequency of the value V_j being classified as category i.

In our PP-attachment problem, the effect of this metric is that words (as feature values) are grouped according to the category distribution of the patterns they belong to. It is possible to cluster the distributions of the values over the categories, and obtain classes of similar words in this fashion. For an example of this type of unsupervised learning as a side-effect of supervised learning, see Daelemans, Berck, and Gillis (1996). In a sense, the MVDM can be interpreted as implicitly implementing a statistically induced, distributed, non-symbolic representation of the words. In this case, the category distribution for a specific word is its lexical representation. Note that the representation for each word is entirely dependent on its behavior with respect to a particular classification task.

In many practical applications of MB-NLP, we are confronted with a very limited set of examples. This poses a serious problem for the MVD metric. Many values occur only once in the whole data set. This means that if two such values occur with the same class, the MVDM will regard them as identical, and if they occur with two different classes their distance will be maximal. In many cases, the latter condition reduces the MVDM to the overlap metric, and additionally some cases will be counted as an exact match on the basis of very shaky evidence. It is, therefore, worthwhile to investigate whether the value difference matrix $\delta(V_i, V_j)$ can be reused from one task to another. This would make it possible to reliably estimate all the δ parameters on a task for which we have a large amount of training material, and to profit from their availability for the MVDM of a smaller domain.

Such a possibility of reuse of lexical similarity is found in the application of Lexical Space representations (Schütze 1994; Zavrel and Veenstra 1995). In LexSpace, each word is represented by a vector of real numbers that stands for a "fingerprint" of the words' distributional behavior across local contexts in a large corpus. The distances between vectors can be taken as a measure of similarity. In Table 1, a number of examples of nearest neighbors are shown.

For each focus-word f, a score is kept of the number of co-occurrences of words from a fixed set of C context-words w_i ($1 < i < C$) in a large corpus. Previous work by Hughes (1994) indicates that the two neighbors on the left and on the right (i.e. the words in positions $n-2$, $n-1$, $n+1$, $n+2$, relative to word n) are a good choice of context. The position of a word in Lexical Space is thus given by a four component vector, of which each component has as many dimensions as there are context words. The dimensions represent the conditional probabilit-

IN	in			
for(in)0.05	since(in)0.10	at(in)0.11	after(in)0.11	under(in)0.11
on(in)0.12	until(in)0.12	by(in)0.13	among(in)0.14	before(in)0.16
GROUP	nn			
network(nn)0.08	firm(nn)0.11	measure(nn)0.11	package(nn)0.11	chain(nn)0.11
club(np)0.11	bill(nn)0.11	partnership(nn)0.12	panel(nn)0.12	fund(nn)0.12
JAPAN	np			
china(np)0.16	france(np)0.16	britain(np)0.19	canada(np)0.19	mexico(np)0.19
india(np)0.19	australia(np)0.20	korea(np)0.22	italy(np)0.23	detroit(np)0.23

Table 1: Some examples of the direct neighbors of words in a Lexical Space (context:250 lexicon:5000 norm:1). The 10 nearest neighbors of the word in upper case are listed by ascending distance.

ies $P(w_1^{n-2}|f) \ldots P(w_c^{n+2}|f)$.

We derived the distributional vectors of all 71479 unique words present in the 3 million words of Wall Street Journal text, taken from the ACL/DCI CD-ROM I (1991). For the contexts, i.e. the dimensions of Lexical Space, we took the 250 most frequent words.

To reduce the 1000 dimensional Lexical Space vectors to a manageable format we applied Principal Component Analysis[1] (PCA) to reduce them to a much lower number of dimensions. PCA accomplishes the dimension reduction that preserves as much of the structure of the original data as possible. Using a measure of the correctness of the classification of a word in Lexical Space with respect to a linguistic categorization (see Zavrel and Veenstra (1995)) we found that PCA can reduce the dimensionality from 1000 to as few as 25 dimensions with virtually no loss, and sometimes even an improvement of the quality of the organization.

Note that the LexSpace representations are task independent in that they only reflect the structure of neighborhood relations between words in text. However, if the task at hand has some positive relation to context prediction, Lexical Space representations are useful.

2 MBL for PP attachment

This section describes experiments with a number of Memory-Based models for PP attachment disambiguation. The first model is based on the lexical information only, i.e. the attachment decision is made by looking only at the identity of the words in the pattern. The second model considers the issue of lexical representation in the MBL framework, by taking as features either task dependent (MVDM) or task independent (LexSpace) syntactic vector representations for words. The introduction of vector representations leads to a number of modifications to the distance metrics and extrapolation rules in the MBL framework. A final experiment examines a number of weighted voting rules.

The experiments in this section are conducted on a simplified version of the "full" PP-attachment problem, i.e. the attachment of a PP in the sequence: `VP`

---

[1] Using the `simplesvd` package, which was kindly provided by Hinrich Schutze. This software can be obtained from `ftp://csli.stanford.edu /pub/prosit/papers/simplesvd/`.

`NP PP`. The data consist of four-tuples of words, extracted from the Wall Street Journal Treebank (Marcus, Santorini, and Marcinkiewicz 1993) by a group at IBM (Ratnaparkhi, Reynar, and Roukos 1994).[2] They took all sentences that contained the pattern `VP NP PP` and extracted the head words from the constituents, yielding a `V N1 P N2` pattern. For each pattern they recorded whether the PP was attached to the verb or to the noun in the treebank parse. Example sentences 1 and 2 would then become:

3 `eats, pizza, with, fork, V.`

4 `eats, pizza, with, anchovies, N.`

The data set contains 20801 training patterns, 3097 test patterns, and an independent validation set of 4039 patterns for parameter optimization. It has been used in statistical disambiguation methods by Ratnaparkhi, Reynar, and Roukos (1994) and Collins and Brooks (1995); this allows a comparison of our models to the methods they tested. All of the models described below were trained on all of the training examples and the results are given for the 3097 test patterns. For the benchmark comparison with other methods from the literature, we use only results for which all parameters have been optimized on the validation set.

In addition to the computational work, Ratnaparkhi, Reynar, and Roukos (1994) performed a study with three human subjects, all experienced treebank annotators, who were given a small random sample of the test sentences (either as four-tuples or as full sentences), and who had to give the same binary decision. The humans, when given the four-tuple, gave the same answer as the Treebank parse 88.2 % of the time, and when given the whole sentence, 93.2 % of the time. As a baseline, we can consider either the Late Closure principle, which always attaches to the noun and yields a score of only 59.0 % correct, or the most likely attachment associated with the preposition, which reaches an accuracy of 72.2 %.

The training data for this task are rather sparse. Of the 3097 test patterns, only 150 (4.8 %) occurred in the training set; 791 (25.5 %) patterns had at least 1 mismatching word with any pattern in the training set; 1963 (63.4 %) patterns at least 2 mismatches; and 193 (6.2 %) patterns at least 3 mismatches. Moreover, the test set contains many words that are not present in any of the patterns in the training set. Table 2 shows the counts of feature values and unknown values. This table also gives the Information Gain estimates of feature relevance.

2.1 Overlap-Based Models

In a first experiment, we used the IB1 algorithm and the IB1-IG algorithm. The results of these algorithms and other methods from the literature are given in Table 3. The addition of IG weights clearly helps, as the high weight of the P feature in effect penalizes the retrieval of patterns which do not match in the preposition. As we have argued in Zavrel and Daelemans (1997), this corresponds exactly to the behavior of the Back-Off algorithm of Collins and Brooks (1995), so that it comes

---

[2] The dataset is available from `ftp://ftp.cis.upenn.edu/pub/adwait/PPattachData/`. We would like to thank Michael Collins for pointing this benchmark out to us.

Feature	train values	total values	unknown	IG weight
V	3243	3475	232	0.03
N1	4315	4613	298	0.03
P	66	69	3	0.10
N2	5451	5781	330	0.03
C	2	2	0	–

Table 2: Statistics of the PP attachment data set.

Method	percent correct
Overlap	83.7 %
Overlap IG ratio	84.1 %
C4.5	79.7 %
Maximum Entropy	77.7 %
Transformations	81.9 %
Back-off model	84.1 %
Late Closure	59.0 %
Most Likely for each P	72.0 %

Table 3: Scores on the Ratnaparkhi et al. PP-attachment test set (see text); the scores of Maximum Entropy are taken from Ratnaparkhi et al. (1994); the scores of Transformations and Back-off are taken from Collins & Brooks (1995). The C4.5 decision tree results, and the baselines have been computed by the authors.

as no surprise that the accuracy of both methods is the same. Note that the Back-Off model was constructed after performing a number of validation experiments on held-out data to determine which terms to include and, more importantly, which to exclude from the back-off sequence. This process is much more laborious than the automatic computation of IG-weights on the training set.

The other methods for which results have been reported on this dataset include decision trees, Maximum Entropy (Ratnaparkhi, Reynar, and Roukos 1994), and Error-Driven Transformation-Based Learning (Brill and Resnik 1994),[3] which were clearly outperformed by both IB1 and IB1-IG, even though e.g. Brill & Resnik used more elaborate feature sets (words and WordNet classes). Adding more elaborate features is also possible in the MBL framework. In this paper, however, we focus on more effective use of the existing features. Because the Overlap metric neglects information about the degree of mismatch if feature-values are not identical, it is worthwhile to look at more finegrained representations and metrics.

---

[3] The results of Brill's method on the present benchmark were reconstructed by Collins and Brooks (1995).

2.2 Continuous Vector Representations for Words

In experiments with Lexical Space representations, every word in a pattern was replaced by its PCA compressed LexSpace vector, yielding patterns with 25x4 numerical features and a discrete target category. The distance metric used was the sum of the LexSpace vector distance per feature, where the distance between two vectors is computed as one minus the cosine, normalized by the cumulative norm. Because no two patterns have the same distance in this case, to use only the nearest neighbor(s) means extrapolating from exactly one nearest neighbor.

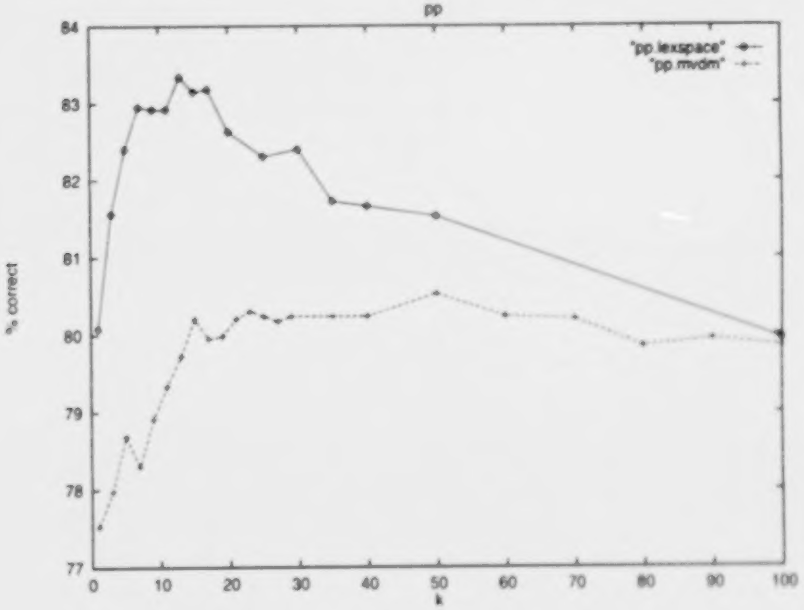


Figure 1: Accuracy on the PP-attachment test set of of MVDM and LexSpace representations as a function of *k*, the number of nearest neighbors.

In preliminary experiments, this was found to give bad results, so we also experimented with various settings for *k* : the parameter that determines the number of neighbors considered for the analogy. The same was done for the MVDM metric which has a similar behavior. We found that LexSpace performed best when *k* was set to 13 (83.3 % correct); MVDM obtained its best score when *k* was set to 50 (80.5 % correct). Although these parameters were found by optimization on the test set, we can see in Figure 1 that LexSpace actually outperforms MVDM for all settings of *k*. Thus, the representations from LexSpace which represent the behavior of the values independent of the requirements of this particular classification task outperform the task specific representations used by MVDM. The reason is that the task specific representations are derived only from the small number of occurrences of each value in the training set, whereas the amount of text available to refine the LexSpace vectors is practically unlimited. Lexical Space however, does not outperform the simple Overlap metric (83.7 % correct) in this form. We suspected that the reason for this is the fact that when continuous representations are used, the number of neighbors is exactly fixed to k, whereas the number of neighbors used in the Overlap metric is, in effect, dependent on the specificity of the match.

2.3 Weighted Voting

This section examines possibilities for improving the behavior of LexSpace vectors for MBL by considering various *weighted voting* methods.

The fixed number of neighbors in the continuous metrics can result in an *oversmoothing* effect. The k-NN classifier tries to estimate the conditional class probabilities from samples in a local region of the data space. The radius of the region is determined by the distance of the k-furthest neighbor. If k is very small and i) the nearest neighbors are not nearby due to data sparseness, or ii) the nearest neighbor classes are unreliable due to noise, the "local" estimate tends to be very poor, as illustrated in Figure 1. Increasing k and thus taking into account a larger region around the query in the dataspace makes it possible to overcome this effect by smoothing the estimate. However, when the majority voting method is used, smoothing can easily become oversmoothing, because the radius of the neighborhood is as large as the distance of the k'th nearest neighbor, irrespective of the local properties of the data. Selected points from beyond the "relevant neighborhood" will receive a weight equal to the close neighbors in the voting function, which can result in unnecessary classification errors.

A solution to this problem is the use of a weighted voting rule which weights the vote of each of the nearest neighbors by a function of their distance to the test pattern (query). This type of voting rule was first proposed by Dudani (1976). In his scheme, the nearest neighbor gets a weight of 1, the furthest neighbor a weight of 0, and the other weights are scaled linearly to the interval in between.

$$w_j = \begin{cases} \frac{d_k - d_j}{d_k - d_1} & \text{if } d_k \neq d_1 \\ 1 & \text{if } d_k = d_1 \end{cases} \tag{6}$$

where d_j is the distance to the query of the j'th nearest neighbor, d_1 the distance of the nearest neighbor, and d_k the distance of the furthest (k'th) neighbor.

Dudani further proposed the *inverse distance weight* (Equation 7), which has recently become popular in the MBL literature (Wettschereck 1994). In Equation 7, a small constant is usually added to the denominator to avoid division by zero.

$$w_j = \frac{1}{d_j} \tag{7}$$

Another weighting function considered here is based on the work of Shepard (1987), who argues for a universal perceptual law, in which the relevance of a previous stimulus for the generalization to a new stimulus is an exponentially decreasing function of its distance in a psychological space. This gives the weighed voting function of Equation 8, where α and β are constants determining the slope and the power of the exponential decay function. In the experiments reported below, $\alpha = 3.0$ and $\beta = 1.0$.

$$w_j = e^{-\alpha d_j^{\beta}} \tag{8}$$

Figure 2 shows the results on the test set for a wide range of k for these voting methods when applied to the LexSpace represented PP-attachment dataset.

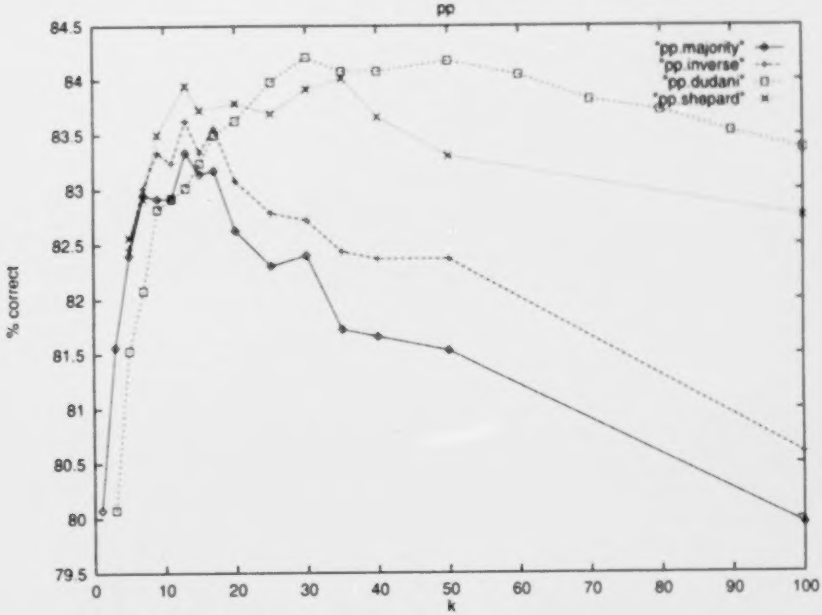


Figure 2: Accuracy on the PP-attachment test set of various voting methods as a function of k, the number of nearest neighbors.

With the inverse distance weighting function the results are better than with majority voting, but here, too, we see a steep drop for k's larger than 17. Using Dudani's weighting function, the results become optimal for larger values of k, and remain good for a wide range of k values. Dudani's weighting function also gives us the best overall result, i.e. if we use the best possible setting for k for each method, as determined by performance on the validation set (see Table 4).

The Dudani weighted k-nearest neighbor classifier (k=30) slightly outperforms Collins & Brooks' (1995) Back-Off model. A further small increase was obtained by combining LexSpace representations with IG weighting of the features, and Dudani's weighted voting function. Although the improvement over Back-Off is

Method	% correct
LexSpace (Dudani, k=30)	84.2 %
LexSpace (Dudani, k=50, IG)	84.4 %

Table 4: Scores on the Ratnaparkhi et al. PP-attachment test set with Lexical Space representations. The values of k, the voting function, and the IG weights were determined on the training and validation sets.

quite limited, these results are nonetheless interesting because they show that MBL can gain from the introduction of extra information sources, whereas this is very difficult in the Back-Off algorithm. For comparison, consider that the performance of the Maximum Entropy model with distributional word-class features is still only 81.6% on this data.

3 Discussion

If we compare the accuracy of humans on the V,N,P,N patterns (88.2 % correct) with that of our most accurate method (84.4 %), we see that the paradigm of learning disambiguation methods from corpus statistics offers good prospects for an effective solution to the problem. After the initial effort by Hindle and Rooth (1993), it has become clear that this area needs statistical methods in which an easy integration of many information sources is possible. A number of methods have been applied to the task with this goal in mind.

Brill and Resnik (1994) applied Error-Driven Transformation-Based Learning to this task, using the *verb*, *noun*1, *preposition*, and *noun*2 features. Their method tries to maximize accuracy with a minimal amount of rules. They found an increase in performance by using semantic information from WordNet. Ratnaparkhi, Reynar, and Roukos (1994) used a Maximum Entropy model and a decision tree on the dataset they extracted from the Wall Street Journal corpus. They also report performance gains with word features derived by an unsupervised clustering method. Ratnaparkhi et al. ignored low frequency events. The accuracy of these two approaches is not optimal. This is most likely due to the fact that they treat low frequency events as noise, though these contain a lot of information in a sparse domain such as PP-attachment. Franz (1996) used a Loglinear model for PP attachment. The features he used were the preposition, the verb level (the lexical association between the *verb* and the *preposition*), the noun level (idem dito for *noun*1), the noun tag (POS-tag for *noun*1), noun definiteness (of *noun*1), and the PP-object tag (POS-tag for *noun*2). A Loglinear model keeps track of the interaction between all the features, though at a fairly high computational cost. The dataset that was used in Franz' work is no longer available, making a direct comparison of the performance impossible. Collins and Brooks (1995) used a Back-Off model, which enables them to take low frequency effects into account on the Ratnaparkhi dataset (with good results). In Zavrel and Daelemans (1997) it is shown that Memory-Based and Back-Off type methods are closely related, which is mirrored in the performance levels. Collins and Brooks got slightly better results (84.5 %) after reducing the sparse data problem by preprocessing the dataset, e.g. replacing all four-digit words with 'YEAR'. The experiments with Lexical Space representations have as yet not shown impressive performance gains over Back-Off, but they have demonstrated that the MBL framework is well-suited to experimentation with rich lexical representations.

4 Conclusion

We have shown that our MBL approach is very competent in solving attachment ambiguities; it achieves better generalization performance than many previous statistical approaches. Moreover, because we can measure the relevance of the features using an information gain metric (IB1-IG), we are able to add features without a high cost in model selection or an explosion in the number of parameters.

An additional advantage of the MBL approach is that, in contrast to the other statistical approaches, it is founded in the use of similarity-based reasoning. Therefore, it makes it possible to experiment with different types of distributed non-symbolic lexical representations extracted from corpora using unsupervised learning. This promises to be a rich source of extra information. We have also shown that task specific similarity metrics such as MVDM are sensitive to the sparse data problem. LexSpace is less sensitive to this problem because of the large amount of data which is available for its training.

Acknowledgements

This research was done in the context of the "Induction of Linguistic Knowledge" research programme, partially supported by the Foundation for Language Speech and Logic (TSL), which is funded by the Netherlands Organization for Scientific Research (NWO).

References

Aha, D., D. Kibler, and M. Albert (1991). Instance-based learning algorithms. *Machine Learning 6*, 37–66.

Brill, E. and P. Resnik (1994). A rule-based approach to prepositional phrase attachment disambiguation. In *Proc. of 15th annual conference on Computational Linguistics*.

Cardie, C. (1996). Automatic feature set selection for case-based learning of linguistic knowledge. In *Proc. of Conference on Empirical Methods in NLP*. University of Pennsylvania.

Collins, M. and J. Brooks (1995). Prepositional phrase attachment through a backed-off model. In *Proc. of Third Workshop on Very Large Corpora*, Cambridge.

Cost, S. and S. Salzberg (1993). A weighted nearest neighbour algorithm for learning with symbolic features. *Machine Learning 10*, 57–78.

Daelemans, W. (1995). Memory-based lexical acquisition and processing. In P. Steffens (Ed.), *Machine Translation and the Lexicon*, Volume 898 of *Lecture Notes in Artificial Intelligence*, pp. 85–98. Berlin: Springer-Verlag.

Daelemans, W. (1996). Abstraction considered harmful: Lazy learning of language processing. In *Proc. of 6th Belgian-Dutch Conference on Machine Learning*, pp. 3–12. Benelearn.

Daelemans, W., P. Berck, and S. Gillis (1996). Unsupervised discovery of phonological categories through supervised learning of morphological rules. In *Proc. of 16th Int. Conf. on Computational Linguistics*, pp. 95–100. Center for Sprogteknologi.

Daelemans, W. and A. van den Bosch (1992). Generalisation performance of backpropagation learning on a syllabification task. In *Proc. of TWLT3: Connectionism and NLP*, pp. 27–37. Twente University.

Dudani, S. (1976). The distance-weighted *k*-nearest neighbor rule. In *IEEE Transactions on Systems, Man, and Cybernetics*, Volume SMC-6, pp. 325–327.

Franz, A. (1996). Learning PP attachment from corpus statistics. In S. Wermter, E. Riloff, and G. Scheler (Eds.), *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, Volume 1040 of *Lecture Notes in Artificial Intelligence*, pp. 188–202. New York: Springer-Verlag.

Frazier, L. (1979). On Comprehending Sentences: Syntactic Parsing Strategies. Ph.d thesis, University of Connecticut.

Hindle, D. and M. Rooth (1993). Structural ambiguity and lexical relations. *Computational Linguistics 19*, 103–120.

Hughes, J. (1994). Automatically Acquiring a Classification of Words. Ph.d thesis, School of Computer Studies, The University of Leeds.

Marcus, M., B. Santorini, and M. Marcinkiewicz (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics 19*, 313–330.

Quinlan, J. (1993). *c4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Ratnaparkhi, A., J. Reynar, and S. Roukos (1994, March). A maximum entropy model for prepositional phrase attachment. In *Workshop on Human Language Technology*, Plainsboro, NJ. ARPA.

Schütze, H. (1994). Distributional part-of-speech tagging. In *Proc. of 7th Conference of the European Chapter of the Association for Computational Linguistics*, Dublin, Ireland.

Shepard, R. (1987). Toward a universal law of generalization for psychological science. *Science 237*, 1317–1228.

Stanfill, C. and D. Waltz (1986, December). Toward memory-based reasoning. *Communications of the ACM 29*(12), 1213–1228.

Wettschereck, D. (1994). A study of distance-based machine learning algorithms. Ph.d thesis, Oregon State University.

Zavrel, J. and W. Daelemans (1997). Memory-based learning: Using similarity for smoothing. In *Proc. of 35th annual meeting of the ACL*, Madrid.

Zavrel, J. and J. Veenstra (1995). The language environment and syntactic word class acquisition. In F. Wijnen and C. Koster (Eds.), *Proc. of Groningen Assembly on Language Acquisition (GALA95)*, Groningen.

Addresses of Authors

Gosse Bouma
Faculty of Arts
University of Groningen
P.O. Box 716
9700 AS Groningen
The Netherlands
gosse@let.rug.nl

Crit Cremers
Department of General Linguistics
Leiden University
P.O. Box 9515
2300 RA Leiden
The Netherlands
cremers@rullet.leidenuniv.nl

Walter Daelemans
Induction of Linguistic Knowledge
Computational Linguistics & AI Group
Tilburg University
P.O. Box 90153
5000 LE Tilburg
The Netherlands
Walter.Daelemans@kub.nl

Maarten Hijzelendoorn
Department of General Linguistics
Leiden University
P.O. Box 9515
2300 RA Leiden
The Netherlands
hijzelendrn@rullet.leidenuniv.nl

Esther Klabbers
IPO, Center for Research on
User-System Interaction
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven
The Netherlands
klabbers@ipo.tue.nl

Dimitra Kolliakou
Dept. of English
Hebrew University of Jerusalem
Mount Scopus
Jerusalem 91905
Israel
dimitra@let.rug.nl

Wessel Kraaij
Netherlands Organization for Applied
Scientific Research (TNO)
Institute of Applied Physics
P.O. Box 155
2600 AD Delft
The Netherlands
kraaij@tpd.tno.nl

Emiel Krahmer
IPO, Center for Research on
User-System Interaction
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven
The Netherlands
krahmer@ipo.tue.nl

Anna Kupść
Polish Academy of Sciences
Institute of Computer Science
21, Ordona
01-237 Warsaw
Poland
aniak@wars.ipipan.waw.pl

Carlos Martín-Vide
Research Group on Mathematical Linguistics and Language Engineering (GRLMC)
Rovira i Virgili University
Pl. Imperial Tàrraco, 1
43005 Tarragona
Spain
cmv@astor.urv.es

Gheorghe Păun
Institute of Mathematics
Romanian Academy
P.O. Box 1-764
70700 București
Romania
gpaun@imar.ro

Renée Pohlmann
Utrecht Institute of Linguistics OTS
Utrecht University
Trans 10
3512 JK Utrecht
The Netherlands
Renee.Pohlmann@let.ruu.nl

Adam Przepiórkowski
Seminar für Sprachwissenschaft
Universität Tübingen
Wilhelmstr. 113
D-72074 Tübingen
Germany
adamp@sfs.nphil.uni-tuebingen.de

Stephen G. Pulman
SRI International Cambridge
Computer Science Research Centre
Suite 23, Miller's Yard
Mill Lane
Cambridge CB2 1RQ
United Kindom
Stephen.Pulman@cam.sri.com

Mieke Rats
Department of Technical Informatics
Delft University of Technology
P.O. Box 356
2600 AJ Delft
The Netherlands
Mieke.Rats@kgs.twi.tudelft.nl

Ineke Schuurman
Centrum voor Computerlinguïstiek
K.U. Leuven
Maria-Theresiastraat 21
3000 Leuven
Belgium
ineke@ccl.kuleuven.ac.be

Mariët Theune
IPO, Center for Research on
User-System Interaction
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven
The Netherlands
theune@ipo.tue.nl

Kees van Deemter
Information Technology Research Institute (ITRI)
University of Brighton
Lewes Road
Brighton BN2 4GJ
United Kingdom
Kees.van.Deemter@itri.brighton.ac.uk

Frank van Eynde
Centrum voor Computerlinguïstiek
K.U. Leuven
Maria-Theresiastraat 21
3000 Leuven
Belgium
frank.vaneynde@ccl.kuleuven.ac.be

Jorn Veenstra
Induction of Linguistic Knowledge
Tilburg University
P.O. Box 90153
5000 LE Tilburg
The Netherlands
jorn.veenstra@kub.nl

Jakub Zavrel
Induction of Linguistic Knowledge
Tilburg University
P.O. Box 90153
5000 LE Tilburg
The Netherlands
Jakub.Zavrel@kub.nl